2011

Novel data clustering methods and applications

Sijia Liu
Iowa State University

Novel data clustering methods and applications

by

Sijia Liu

A dissertation submitted to the graduate faculty

in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Major: Applied Mathematics

Program of Study Committee:

Anastasios Matzavinos, Major Professor

Paul Sacks

Jim W. Evans

Sunder Sethuraman

Alexander Roitershtein

Iowa State University

Ames, Iowa

2011

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGEMENTS

It is my pleasure to express my gratitude to my advisor, Dr. Anastasios Matzavinos, for his continuous support, motivation, encouragement, and help during my Ph.D. study and research. Without his help, this thesis would not be possible. With his guidance, I started to learn how to do research and how to grow academically with collaborative spirit. With his motivation and encouragement, I started to explore the strength of innovation.

I would like to thank Dr. Sunder Sethuraman, with whom I have collaborated on the research of data clustering. The collaboration experience showed me the joint strength of theoretical and problem driven research. I want to give my gratitude to Dr. Alexander Roitershtein. It has been an educational and interesting experience to collaborate with him. I am also grateful to all my other collaborators for their help.

I owe my gratitude to Dr. Paul Sacks, Dr. James Evans, and Dr. Howard Levine, who have given me generous help and encouragement. Dr. Sacks has been overseeing my study and research as the Director of Graduate Education in Department of Mathematics, and has given invaluable guidance and support in critical situations. Dr. Levine has trusted my ability, encouraged me and kept me motivated to overcome difficulties. Dr. Evans has shown me a broad view of applications of mathematics in physics and chemistry.

I am given the opportunity to pursue the Ph.D. degree by the Department of Mathematics, Iowa State University, to whom I owe my sincere gratitude. I would like to thank all my friends for supporting me and spending the five years with me.

I owe my sincere gratitude to Dominic Kramer, who always understands me, believes in

me, and supports me. Without his help, I could not be able to focus almost all my energy to solve challenging problems, to regenerate from setbacks, and to be optimistic even when I am facing difficulties.

I want to thank my family, especially my parents, Bianxia Duan and Bing Liu, for giving birth to me, for nurturing and educating me, for having faith in me, and for their constant care and support throughout my life. My grandfather, Jiantang Liang, has been educating me since I was young. Last but not least, I would like to give special thanks to my grandmother, Xiulan Xia, who had the kindest heart and will always be with me in my heart.

ABSTRACT

The need to interpret and extract possible inferences from high-dimensional datasets has led over the past decades to the development of dimensionality reduction and data clustering techniques. Scientific and technological applications of clustering methodologies include among others bioinformatics, biomedical image analysis and biological data mining. Current research in data clustering focuses on identifying and exploiting information on dataset geometry and on developing robust algorithms for noisy datasets. Recent approaches based on spectral graph theory have been devised to efficiently handle dataset geometries exhibiting a manifold structure, and fuzzy clustering methods have been developed that assign cluster membership probabilities to data that cannot be readily assigned to a specific cluster.

In this thesis, we develop a family of new data clustering algorithms that combine the strengths of existing spectral approaches to clustering with various desirable properties of fuzzy methods. More precisely, we consider a slate of "random-walk" distances arising in the context of several weighted graphs formed from the data set, which allow to assign "fuzzy" variables to data points which respect in many ways their geometry. The developed methodology groups together data which are in a sense "well-connected", as in spectral clustering, but also assigns to them membership values as in other commonly used fuzzy clustering approaches. This approach is very well suited for image analysis applications and, in particular, we use it to develop a novel facial recognition system that outperforms other well-established methods.

CHAPTER 1. Introduction

The need to interpret and extract possible inferences from high-dimensional datasets has led over the past decades to the development of dimensionality reduction and data clustering techniques Filippone et al. (2007). Scientific and technological applications of such clustering methodologies include among others computer imaging Archip et al. (2005) Shi and Malik (2000), data mining and bioinformatics Liao et al. (2009) Snel et al. (2002). One of the most widely used and studied statistical methods for data clustering is the K-means algorithm, which was first introduced in MacQueen (1967) and is still in use nowadays as the prototypical example of a non-overlapping clustering approach Kogan (2007). The applicability of the K-means algorithm, however, is restricted by the requirement that the clusters to be identified should be well-separated and of a regular, convex-shaped geometry, a requirement that is often not met in practice. In this context, two fundamentally distinct approaches have been proposed in the past to address these restrictions.

Bezdek *et al.* Bezdek et al. (1984) proposed the fuzzy c-means (FCM) algorithm as an alternative, soft clustering approach that generates fuzzy partitions for a given dataset. In the case of FCM the clusters to be identified do not have to be well-separated, as the method assigns cluster membership probabilities to undecidable elements of the dataset that cannot be readily assigned to a specific cluster. However, the method does not exploit the intrinsic geometry of non-convex clusters, and its performance is drastically reduced when applied to datasets that are curved, elongated or contain clusters of different dispersion. This behaviour can also be observed in the case of the standard K-means algorithm Ng et al. (2001). Although these algorithms have been successful in a number of examples, this thesis focuses on datasets for which their performance is poor.

More recently, approaches based on spectral graph theory have been devised to circumvent

the computational problems associated with the geometry of datasets exhibiting a manifold structure. Such approaches exploit the information encoded in the spectrum of specific linear operators acting on the data, such as the normalized graph Laplacian. The effectiveness of such spectral methodologies has been attributed to the analogy that exists between the graph Laplacian and the Beltrami operator Rosenberg (1997). The latter is defined in Chapter 3 and it provides the means for describing the process of diffusion on Riemannian manifolds and can consequently be used to identify the optimal embedding dimension, which in turn has natural connections to clustering Belkin and Niyogi (2003). As discussed in Chapters 3 and 4, an alternative analysis focuses on the properties of the normalized graph Laplacian as a stochastic matrix. The eigenvalues of the latter represent the various time scales of the corresponding random walk, and this information can be used to identify efficient representations of the underlying dataset, hence facilitating the process of clustering Coifman and Lafon (2006). In the following, we define the basic mathematical notions underlying clustering methods, such as the ones described above, and delineate the contents of this thesis.

1.1 Centroid-based clustering methods

We introduce here some of the basic notions underlying the classical k-means and fuzzy c-means methods. More details are provided in Chapter 2, and the spectral clustering approach is introduced in Chapter 3. In what follows, we consider a set of data

$$\mathcal{D} = \{x_1, x_2, \ldots, x_n\} \subset \mathbb{R}^m.$$

embedded in a Euclidean space. The output of a data clustering algorithm is a partition:

$$\Pi = \{\pi_1, \pi_2, \ldots, \pi_k\}, \tag{1.1}$$

where $k \leq n$ and each π_i is a nonempty subset of $\mathcal{D}$. Π is a partition of $\mathcal{D}$ in the sense that

$$\bigcup_{i \leq k} \pi_i = \mathcal{D} \text{ and } \pi_i \cap \pi_j = \varnothing \text{ for all } i \neq j. \tag{1.2}$$

In this context, the elements of Π are usually referred to as clusters. In practice, one is interested in partitions of $\mathcal{D}$ that satisfy specific requirements, usually expressed in terms of a distance function $d(\cdot, \cdot)$ that is defined on the background Euclidean space.

The classical k-means algorithm is based on reducing the notion of a cluster π_i to that of a cluster representative or centroid $c(\pi_i)$ according to the relation

$$c(\pi_i) = \arg\min \left\{ \sum_{x \in \pi_i} d(x, y) \;\middle|\; y \in \mathbb{R}^m \right\}. \tag{1.3}$$

In its simplest form, k-means consists of initializing a random partition of $\mathcal{D}$ and subsequently updating iteratively the partition Π and the centroids $\{c(\pi_i)\}_{i \leq k}$ through the following two steps (see, e.g., Kogan (2007)):

(a) Given $\{\pi_i\}_{i \leq k}$, update $\{c(\pi_i)\}_{i \leq k}$ according to (1.3).

(b) Given $\{c(\pi_i)\}_{i \leq k}$, update $\{\pi_i\}_{i \leq k}$ according to centroid proximity, i.e., for each $i \leq k$,

$$\pi_i = \left\{ x \in \mathcal{D} \mid d(c_i, x) \leq d(c_j, x) \text{ for each } j \leq k \right\}$$

In applications, it is often desirable to relax condition (1.2) in order to accommodate for overlapping clusters Fu and Medico (2007). Moreover, condition (1.2) can be too restrictive in the context of filtering data outliers that are not associated with any of the clusters present in the data set. These restrictions are overcome by fuzzy clustering approaches that allow the determination of outliers in the data and accommodate multiple membership of data to different clusters Ma; and Wu (2007).

In order to introduce fuzzy clustering algorithms, we reformulate condition (1.2) as:

$$u_{ij} \in \{0, 1\}, \;\; \sum_{\ell=1}^{k} u_{\ell j} = 1, \;\text{ and }\; \sum_{\ell=1}^{n} u_{i\ell} > 0, \tag{1.4}$$

for all $i \leq k$ and $j \leq n$, where u_{ij} denotes the membership of datum x_j to cluster π_i (i.e., $u_{ij} = 1$ if $x_j \in \pi_i$, and $u_{ij} = 0$ if $x_j \notin \pi_i$). The matrix $(u_{ij})_{i \leq k, j \leq n}$ is usually referred to as the data membership matrix. In fuzzy clustering approaches, u_{ij} is allowed to range in the interval $[0, 1]$ and condition (1.4) is replaced by:

$$u_{ij} \in [0, 1], \;\; \sum_{\ell=1}^{k} u_{\ell j} = 1, \;\text{ and }\; \sum_{\ell=1}^{n} u_{i\ell} > 0, \tag{1.5}$$

for all $i \leq k$ and $j \leq n$ Bezdek et al. (1984). In light of Eq. (1.5), the matrix $(u_{ij})_{i \leq k, j \leq n}$ is sometimes referred to as a fuzzy partition matrix of $\mathcal{D}$. For each $j \leq n$, $\{u_{ij}\}_{i \leq k}$ defines

a probability distribution with u_{ij} denoting the probability of data point x_j being associated with cluster π_i. Hence, fuzzy clustering approaches are characterized by a shift in emphasis from defining clusters and assigning data points to them to that of a membership probability distribution.

The prototypical example of a fuzzy clustering algorithm is the fuzzy c-means method (FCM) developed by Bezdek *et al.* Bezdek et al. (1984). The FCM algorithm can be formulated as an optimization method for the objective function J_p, given by:

$$J_p(U, C) = \sum_{i=1}^{k} \sum_{j=1}^{n} u_{ij}^p \|x_j - c_i\|^2, \tag{1.6}$$

where $U = (u_{ij})_{i \leq k, j \leq n}$ is a fuzzy partition matrix, i.e. its entries satisfy condition (1.5), and $C = (c_i)_{i \leq k}$ is the matrix of cluster centroids $c_i \in \mathbb{R}^m$. The real number p is a "fuzzification" parameter weighting the contribution of the membership probabilities to J_p Bezdek et al. (1984). In general, depending on the specific application and the nature of the data, a number of different choices can be made on the norm $\|\cdot\|$. The FCM approach consists of globally minimizing J_p for some $p > 1$ over the set of fuzzy partition matrices U and cluster centroids C. The minimization procedure that is usually employed in this context involves an alternating directions scheme Ma; and Wu (2007), which is commonly referred to as the FCM algorithm. A listing of the FCM algorithm is given in Chapter 2.

This approach, albeit conceptually simple, works remarkably well in identifying clusters, the convex hulls of which do not intersect Jain (2008) Meila (2006), with a representative example being discussed in Chapter 2. However, for general data sets, J_p is not convex and, as we demonstrate in Chapter 3, one can readily construct data sets $\mathcal{D}$ for which the standard FCM algorithm fails to detect the global minimum of J_p Ng et al. (2001).

1.2 Outline of the thesis

In this thesis, we consider a slate of "random-walk" distances arising in the context of several weighted graphs formed from the data set, in a comprehensive generalized FCM framework, which allow to assign "fuzzy" variables to data points which respect in many ways their geometry. The method we present groups together data which are in a sense "well-connected",

as in spectral clustering, but also assigns to them membership values as in FCM. We remark our technique is different than say clustering by spectral methods, and then applying FCM. It is also much different than other recent approaches in the literature, such as the FLAME Fu and Medico (2007) and DIFFUZZY Cominetti et al. (2010) algorithms, which compute "core clusters" and try to assign data points to them.

The particular random walk distance focused upon in the thesis, among others, is the "absorption" distance, which is new to the literature (see Chapter 4 for definitions). We remark, however, a few years ago a "commute-time" random walk distance was introduced and used in terms of clustering Yen et al. (2005). In a sense, although our technique is more general and works much differently than the approach in Yen et al. (2005), our method builds upon the work in Yen et al. (2005) in terms of using a random walk distance. In particular, in Chapter 4, we introduce novelties, such as motivated "penalty terms" and "locally adaptive" weights to construct underlying graphs from the given data set, which make Fuzzy-RW impervious to random seed initializations.

The outline of the thesis is the following. First, in Chapter 2, we further discuss the classical FCM algorithm, and delineate some of its merits and demerits with respect to some data sets, including some applications to image segmentation and other image analysis tasks. In Chapter 3, we review the various existing approaches to spectral clustering and machine learning approaches. Chapters 2 and 3 culminate in the development of a family of new clustering methods, dubbed *Fuzzy-RW*, in Chapter 4 which is one of the two main novelties of this thesis. We demonstrate the effectiveness and robustness of Fuzzy-RW on several standard synthetic benchmarks and other standard data sets such as the IRIS and the YALE face data sets Georghiades et al. (2000) in Chapters 4 and 5. In particular, in Chapter 5 we turn our focus on an important application of data clustering, namely the problem of face recognition. After reviewing current approaches and methodologies, we demonstrate that the performance of Fuzzy-RW in face identification is comparable to, and often better than, the performance of existing methods.

CHAPTER 2. K-Means Method and Fuzzy C-Means Method

2.1 Introduction

The K-means method is considered as one of the most popular and standard clustering methods. Although it was initially discovered more than 50 years ago Steinhaus (1956) Lloyd (1982) Ball and Hall (1965) MacQueen (1967) and since then there has been enormous improved techniques designed targeting a variety of applications (see Jain (2008) for a review), the K-means method is still generally applied because of its simplicity and efficiency. The goal of K-means is to partition a given dataset such that data points in a same cluster are similar and the data points in different clusters are dissimilar. It also requires that the clusters are non-overlapping. The clustering algorithms that satisfy the above requirements are classified as crisp data clustering methods. On the other hand, algorithms that allow every data point to be assigned to more than one clusters are classified as fuzzy clustering methods. Fuzzy c-means method, proposed in Dunn (1973) and improved in Bezdek (1981), is a fuzzy clustering method that is analogous to the K-means method. The K-means method and fuzzy c-means method can be varied or improved by being applied with different choices of distance measures Mao and Jain (1996) Linde et al. (1980) Banerjee et al. (2005b), and can be combined with other clustering techniques, for example, kernel methods Zhang et al. (2003), to achieve better results according to the nature of clustering problems. In this chapter, we discuss the K-means method, fuzzy c-means method and their related variants. Section 1 describes the K-means method and fuzzy c-means method. Section 2 discusses the relationship between the K-means method and the principal component analysis. Finally, section 3 describes several extended algorithms based on or related to the K-means and fuzzy c-means method.

2.2 K-Means Method and Fuzzy C-Means Method

For a given dataset $X = \{\mathbf{x}_j\}_{j=1}^{N}$ to be partitioned into K non-overlapping clusters $C_1, C_2, \cdots, C_K$, K-means method aims at finding an optimal partition such that the total distance between data points and the centroids of the clusters to which they are partitioned is minimized. Such a partition is found by minimizing the following objective function:

$$J = \sum_{i=1}^{K} \sum_{\mathbf{x}_j \in C_i} \|\mathbf{x}_j - \mathbf{m}_i\|^2 \tag{2.1}$$

where $\|\cdot\|$ represents the measure of distance, it is proposed as the Euclidean distance in the K-means method. $\mathbf{m}_i$ is the centroid of the i-th cluster.

However, the problem of finding the globally minimized objective function J is NP-hard Drineas et al. (2004). Instead, a partition can be generated by an iterative algorithm. This algorithm can be initiated with a random partition of X, then it iterates between calculating the centroids of the current partition and updating the partition by assigning data points to their nearest centroids. The K-means algorithm can be summarized in Table 2.1.

The above algorithm can only lead to a local minimum of J, thus the clustering result of the K-means method depends on the partition used for initiation. To reduce the influence of the initial partition on the clustering result, one can run the algorithm multiple times using different randomly generated initial partitions and then choose the result that has the smallest total squared distance Jain (2008). See Fig. 2.1 for an example of applying K-means method to cluster a dataset consisting of four groups of data points that have Gaussian distributions.

Fuzzy c-means method, on the other hand, approaches the problem of clustering X into K clusters by assigning each data point membership values as the possibilities for it to be classified into multiple clusters. It requires to minimize the total weighted distance between data points and all centroids, where the weights are the corresponding membership values. The objective

Table 2.1: Algorithm of the K-means method

Input: Dataset $X = \{\mathbf{x}_j\}_{j=1}^N$, the number of clusters K, and a small positive value ϵ or the maximum iteration time T.

Output: A partition $C_1, C_2, \cdots, C_K$ that satisfies $\cup_{i=1}^K C_i = X$ and $C_i \cap C_j = \emptyset$, $\forall i \neq j$.

Initiation: Randomly generate a partition $\{C_i^{(0)}\}_{i=1}^K$ of X.

Iteration:

1. For $i \in \{1, 2, \cdots, K\}$, compute the centroid $\mathbf{m}_i^{(t)}$ of the i-th cluster $C_i^{(t)}$. Compute the objective function

$$J^{(t)} = \sum_{i=1}^{K} \sum_{\mathbf{x}_j \in C_i^{(t)}} \|\mathbf{x}_j - \mathbf{m}_i^{(t)}\|^2$$

2. For a data point $\mathbf{x}_j$, find its nearest centroid $\mathbf{m}_{j^*}^{(t)}$, i.e.

$$\mathbf{m}_{j^*}^{(t)} = \arg\min\{\|\mathbf{x}_j - \mathbf{m}_i^{(t)}\| \quad : \quad \mathbf{m}_i^{(t)} \in \{\mathbf{m}_i^{(t)}\}_{i=1}^K\}$$

   Assign $\mathbf{x}_j$ into the j^*-th cluster for all $j \in \{1, 2, \cdots, N\}$ to form a new partition $\{C_i^{(t+1)}\}_{i=1}^K$.

3. Repeat the above steps 1) and 2) until $\|J^{(t+1)} - J^{(t)}\| < \epsilon$ or the iteration time exceeds T.

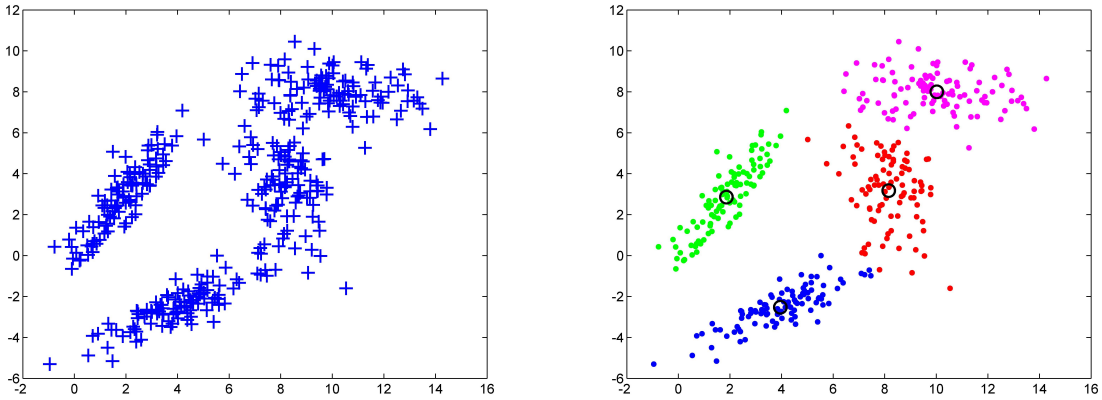


Figure 2.1: (a). A dataset that consists of four groups of data points. Each group of data points are of Gaussian distribution. (b). The clustering result by applying K-means method and requiring the number of clusters to be four. Different clusters are colored with red, blue, green, and magenta. Centroids of clusters are marked with black circles.

function that fuzzy c-means tries to minimize is the following:

$$J_m = \sum_{i=1}^{C} \sum_{j=1}^{N} U_{ij}^m \|\mathbf{x}_j - \mathbf{m}_i\|^2 \tag{2.2}$$

where U_{ij} is the membership value of data point $\mathbf{x}_j$ to be assigned to the i-th cluster, which satisfies the requirements

$$U_{ij} \in [0,1],\ \forall i,\ j,\ \text{and} \quad \sum_{i=1}^{K} U_{ij} = 1,\ \ \forall j \in \{1, \cdots, N\} \tag{2.3}$$

Also, m is the degree of fuzzification Bezdek et al. (2005) that adjusts the effect of the membership values, $1 \leq m < \infty$. Usually m is taken as 2 for simplicity. Fuzzy c-means can be considered as a generalization of the K-means method in the sense that, if we assign each data point into only one cluster and $U_{ij} = 1$ if and only if $\mathbf{x}_j$ is assigned to the i-th cluster, the above J_m is the same as the objective function J in the K-means method.

Similar to the K-means method, an iterative algorithm derived by using the method of Lagrange multiplier can be applied to find a local minimization of J_m for a given a random initiation. The algorithm can be summarized in Table 3.1.

The measure of distances used in the objective functions of the K-means method and the fuzzy c-means method are not restricted to the Euclidean distance. For example, the K-means method has been combined with the Itakura-Saito distance Linde et al. (1980), a family of Bregman distance Banerjee et al. (2005b). And the fuzzy c-means method has been combined with the kernel methods to cluster datasets that have complex nonlinear structures Zhang et al. (2003).

2.3 K-Means Method and Principal Component Analysis

Principal component analysis (PCA) is a standard unsupervised dimension reduction method that has been broadly used, while K-means method is a popular and efficient unsupervised clustering method. It is proved that the principal components obtained from PCA are the contin-

Table 2.2: Algorithm of the fuzzy c-means method

Input Dataset $X = \{\mathbf{x}_j\}_{j=1}^N$, the number of clusters K, and a small positive value ϵ or the maximum iteration time T.

Output A membership matrix U, where U_{ij} represents the probability to assign $\mathbf{x}_j$ to the i-th cluster.

Initiation Randomly generate a membership matrix U^0 that satisfies requirements in (2.3).

Iteration

1. For $i \in \{1, 2, \cdots, K\}$, compute the centroid $\mathbf{m}_i^{(t)}$ of the i-th cluster:

$$\mathbf{m}_i = \frac{\sum_{j=1}^N U_{ij}^m \mathbf{x}_j}{\sum_{j=1}^N U_{ij}^m} \tag{2.4}$$

   Compute the objective function $J_m^{(t)}$ as in (2.2).

2. Update the membership matrix:

$$U_{ij} = \left(\sum_{k=1}^K \left(\frac{\|\mathbf{x}_j - \mathbf{m}_i\|}{\|\mathbf{x}_j - \mathbf{m}_k\|} \right)^{\frac{2}{m-1}} \right)^{-1} \tag{2.5}$$

3. Repeat steps 1) and 2) until $\|J_m^{(t+1)} - J_m^{(t)}\| < \epsilon$ or the iteration time exceeds T.

uous analog [1] of the discrete cluster membership indicators for the K-means method Ding and He (2004). Also, the authors in Ding and He (2004) have showed that the subspace spanned by the cluster centroids derived from K-means method can be represented using the principal components of the data covariance matrix. These justify the effectiveness of PCA-based data reduction as a preprocessor of centroid-based clustering methods. Further, following the similar argument, components of Kernel PCA provides continuous solutions to the Kernel K-means method. Here we follow Ding and He (2004) to show the above relationship between PCA and K-means method.

First, some standard notations for PCA are given here. For a given dataset $X = \{\mathbf{x}_i\}_{i=1}^N$, we consider the centered version $Y = (\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_N)$ where $\mathbf{y}_i = \mathbf{x}_i - \mathbf{m}$ and $\mathbf{m} = \sum_{i=1}^N \mathbf{x}_i/N$. The covariance matrix (without the factor $1/N$) is YY^T. The so called principal directions $\mathbf{u}_k$ and principal components $\mathbf{v}_k$ are vectors satisfy

$$YY^T\mathbf{u}_k = \lambda_k\mathbf{u}_k, \;\; Y^TY\mathbf{v}_k = \lambda_k\mathbf{v}_k, \;\; \mathbf{v}_k = Y^T\mathbf{u}_k/\lambda_k^{1/2} \tag{2.6}$$

And the singular value decomposition of Y is

$$Y = \sum_{k=1}^K \lambda_k^{1/2}\mathbf{u}_k\mathbf{v}_k^T$$

On the other hand, let us consider the K-means method. For a given dataset $X = \{x_i\}_{i=1}^N$ and a predefined number of clusters K, K-means method generates the partition $C_1, C_2, \cdots, C_K$ of X by minimizing the following objective function

$$J_K = \sum_{k=1}^K \sum_{x_i \in C_k} \|x_i - m_k\|^2 \tag{2.7}$$

where m_k is the centroid of the cluster C_k, $m_k = \sum_{x_i \in C_k} x_i/n_k$ with n_k being the number of data points in cluster C_k.

[1] In Ding and He (2004), the authors focus on the solutions that adopt continuous real values instead of discrete values 0, 1.

The above objective function can be transformed as in the following calculation:

$$\begin{aligned}
J_k &= \sum_{k=1}^{K} \sum_{x_i \in C_k} \|x_i - m_k\|^2 = \sum_{k=1}^{K} \sum_{x_i \in C_k} \|x_i - \frac{\sum_{x_j \in C_k} x_j}{n_k}\|^2 \\
&= \sum_{k=1}^{K} \sum_{x_i \in C_k} \left(\|x_i\|^2 - 2\frac{\langle x_i, \sum_{x_j \in C_k} x_j \rangle}{n_k} + \frac{(\sum_{x_j \in C_k} x_j)^2}{n_k^2} \right) \\
&= \sum_{k=1}^{K} \left(\sum_{x_i \in C_k} \|x_i\|^2 - \frac{2}{n_k} \sum_{x_i, x_j \in C_k} \langle x_i, x_j \rangle + \frac{n_k}{n_k^2} \sum_{x_j, x_m \in C_k} \langle x_j, x_m \rangle \right) \\
&= \sum_{k=1}^{K} \frac{1}{n_k} \sum_{x_i, x_j \in C_k} \left(\|x_i\|^2 - \langle x_i, x_j \rangle \right) \\
&= \sum_{k=1}^{K} \sum_{x_i, x_j \in C_k} \frac{1}{2n_k} \|x_i - x_j\|^2
\end{aligned}$$

And the above can be further rewritten as:

$$J_k = \sum_{x_i \in X} \|x_i\|^2 - \sum_{k=1}^{K} \frac{1}{n_k} \sum_{x_i, x_j \in C_k} \langle x_i, x_j \rangle \tag{2.8}$$

If we introduce the indicator matrix $H_k = (\mathbf{h}_1, \cdots, \mathbf{h}_K)$ whose columns are the l_2 unit indicator vectors of the K clusters, i.e. $\mathbf{h}_k \in \mathbb{R}^N$ and

$$\mathbf{h}_k(i) = \frac{1}{n_k^{1/2}} \begin{cases} 1 & \text{if } x_i \in C_k \\ 0 & \text{else} \end{cases} \tag{2.9}$$

Without loss of generality, we can assume that the data points assigned in the same cluster have adjacent indices. Then (2.9) can be expressed using the indicator matrix:

$$J_K = Tr(X^T X) - Tr(H_k^T X^T X H_k) \tag{2.10}$$

where $Tr(H_k^T X^T X H_k) = \mathbf{h}_1^T X^T X \mathbf{h}_1 + \cdots + \mathbf{h}_K^T X^T X \mathbf{h}_K$. Since the $\mathbf{h}_k$'s have relationship $\sum_{k=1}^{K} n_k^{1/2} \mathbf{h}_k = \mathbf{e}$, where $\mathbf{e} \in \mathbf{R}^N$ is the vector of all ones, we can apply a linear transformation on H_k to derive $\mathbf{q}_k$'s:

$$Q_K = (\mathbf{q}_1, \cdots, \mathbf{q}_K) = H_K T, \quad \text{or } \mathbf{q}_l = \sum_k \mathbf{h}_k t_{kl} \tag{2.11}$$

where T is the $K \times K$ transformation matrix that satisfies $T^T T = I$ and we can require that the last column of T is

$$\mathbf{t}_K = (\sqrt{n_1/N}, \cdots, \sqrt{n_K/N})^T \tag{2.12}$$

thus the last column of Q_K is

$$\mathbf{q}_K = \sqrt{\frac{n_1}{N}}\mathbf{h}_1 + \cdots + \sqrt{\frac{n_K}{N}}\mathbf{h}_K = \sqrt{\frac{1}{N}}\mathbf{e} \tag{2.13}$$

Such type of transformation is proved to be always possible Ding and He (2004). The orthogonality of $\mathbf{h}_k$'s imply the same for $\mathbf{q}_k$'s, i.e. since $\mathbf{h}_i^T\mathbf{h}_j = \delta_{ij}$, then

$$\mathbf{q}_i^T\mathbf{q}_j = \sum_p \mathbf{h}_p^T t_{pi} \sum_q \mathbf{h}_q t_{qj} = \sum_{p,q} \mathbf{h}_p^T t_{pi}\mathbf{h}_q t_{qj} = (T^T T)_{ij} = \delta_{ij}$$

Thus, considering (2.13), we can rewrite the orthogonality $Q_K^T Q_K = I$ as the following:

$$Q_{K-1}^T Q_{K-1} = I_{K-1} \tag{2.14}$$

$$\mathbf{q}_k^T\mathbf{e} = 0, \quad \text{for } k = 1, \cdots, K-1 \tag{2.15}$$

Then the objective function of K-means method can be represented by the the dataset and the first $K-1$ columns of Q_K:

$$J_K = Tr(X^T X) - \mathbf{e}^T X^T X\mathbf{e}/N - Tr(Q_{K-1}^T X^T X Q_{k-1}) \tag{2.16}$$

We can also use the centered dataset $Y = \{y_i\}_{i=1}^N$ to represent J_K, where $y_i = x_i - \sum_{x_i \in X} x_i/N$. In terms of Y, the objective function becomes:

$$J_K = Tr(Y^T Y) - Tr(Q_{K-1}^T Y^T Y Q_{K-1}) \tag{2.17}$$

where we used the fact that $Y\mathbf{e} = 0$.

Then the minimization of J_K becomes

$$\max_{Q_{K-1}} Tr(Q_{K-1}^T Y^T Y Q_{K-1}) \tag{2.18}$$

subject to constraints as in (2.14), (2.15) and the fact that every $\mathbf{q}_k$'s are the linear transformation of $\mathbf{h}_k$'s. If only the last constraint is ignored such that $\mathbf{q}_k$'s can take continuous values while satisfying (2.14) and (2.15), then the maximization problem has closed form solution and J_K can be bounded based on the following theorem:

Theorem 1. *The continuous solutions for the transformed discrete cluster indicators of K-means method are the $K-1$ principal components of the dataset X that correspond to the biggest $K-1$ nonzero eigenvalues. J_K satisfies the upper and lower bounds*

$$N\bar{\mathbf{y}^2} - \sum_{k=1}^{K-1} \lambda_k < J_K < N\bar{\mathbf{y}^2} \tag{2.19}$$

where $N\bar{\mathbf{y}^2}$ is the total variance and λ_k is the k-th largest nonzero principal eigenvalue of the covariance matrix Y^TY.

We can check that the first $K-1$ principal components satisfy the constraint in (2.14) because of their mutual orthogonality, they also satisfy (2.15) because $\mathbf{e}$ is the eigenvector corresponding to eigenvalue 0. The proof of this theorem is the direct application of the conclusion in the theorem of Ky Fan Fan (1949).

The relationship between PCA and the K-means method can also be seen when one considers the subspace spanned by the centroids. For the given dataset X that partitioned into K clusters with $\mathbf{m}_k$ as the centroid of the k-th cluster, the between-cluster scatter matrix $S_b = \sum_{k=1}^{K} n_k \mathbf{m}_k \mathbf{m}_k^T$ can be considered as an operator that projects any vector $\mathbf{x}$ into the subspace spanned by the K centroids. That is,

$$S_b^T \mathbf{x} = \sum_{k=1}^{K} n_k \mathbf{m}_k (\mathbf{m}_k^T \mathbf{x})$$

This subspace is called cluster centroid subspace in Cleuziou (2008) and the following theorem reveals its connection to the PCA dimension reduction:

Theorem 2. *Cluster centroid subspace is spanned by the first $K-1$ principal directions of the centered dataset Y. The principal directions $\mathbf{u}_k$ ($k = 1, 2, \cdots, K-1$) are those vectors obtained from the singular value decomposition of Y and they satisfy*

$$YY^T \mathbf{u}_k = \lambda_k \mathbf{u}_k$$

Proof. Consider the centered dataset Y. The centroid of the k-th cluster C_k can be represented using the cluster indicator vector h_k:

$$\mathbf{m}_k = \frac{\sum_{\mathbf{y}_i \in C_k} \mathbf{y}_i}{n_k} = n_k^{-1/2} \sum_i h_k(i) \mathbf{y}_i = n_k^{-1/2} Y h_k$$

Then the between-cluster scatter matrix becomes

$$S_b = \sum_{k=1}^{K} Y h_k h_k^T Y^T = Y \left(\sum_{k=1}^{K} h_k h_k^T \right) Y^T = Y \left(\sum_{k=1}^{K} q_k q_k^T \right) Y^T$$

By Theorem 1, $q_1, \cdots, q_{K-1}$ are linear transformations of $v_1, \cdots, v_{K-1}$, and q_K is $\mathbf{e}/N^{1/2}$. Thus

$$\sum_{k=1}^{K} q_k q_k^T = \mathbf{e}\mathbf{e}^T/N + \sum_{k=1}^{K-1} v_k v_k^T$$

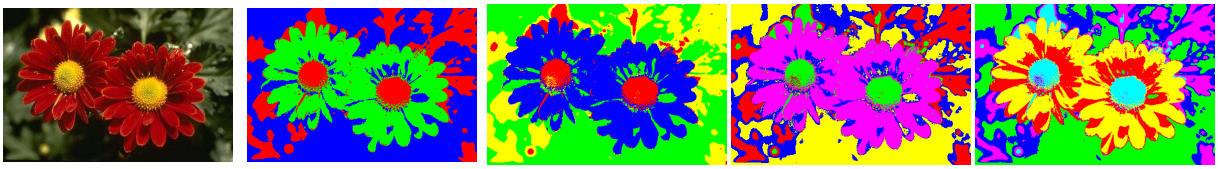
Notice that $Y\mathbf{e} = 0$. By (2.6), we have $Y\mathbf{v}_k = \lambda_k^{1/2}\mathbf{u}_k$, which completes the proof. □

This theorem support the effectiveness of PCA as a preprocessor of clustering methods

2.4 Extensions and Variants of K-Means Method

2.4.1 Kernel Based Fuzzy Clustering with Spatial Constraints

Among the applications of data clustering, the problem of color segmentation of a given image is one that requires clustering the pixels of the digital image based on their colors. Fig. 2.2 shows the color segmentation results by applying fuzzy c-means method on a digital image with specified number of clusters as three, four, five and six respectively.



(a) Original image (b) 3 clusters (c) 4 clusters (d) 5 clusters (e) 6 clusters

Figure 2.2: (a) The original image, (b-e) Color segmentation results by applying fuzzy c-means method. In each result, the assignments of pixels are based on their maximum membership values. The number of clusters in the results are required to be $K = 3$, $K = 4$, $K = 5$, and $K = 6$ respectively.

Although the K-means method and fuzzy c-means method have been applied with empirical success in solving color segmentation problems, their performance are not reliable when the images are corrupted with noise or when there exists irreducible intensity inhomogeneity properties that are induced from the methods of collecting the images, for example, the radio-frequency coil in magnetic resonance imaging (MRI) Zhang et al. (2003). To avoid the case

where the color of a single pixel can affect its assignment in the clustering result dramatically, especially when considering that this pixel may be the noise of a image, methods that take into account of spatial information have been developed under the framework of the fuzzy c-means method Ahmed et al. (2002) Liew et al. (2000) qiang Zhang and can Chen (2004), where Ahmed et al. (2002) qiang Zhang and can Chen (2004) modified the objective function of the fuzzy c-means method directly to incorporate the spatial influence. At the same time, more sophisticated measures of distance can be applied in solving color segmentation problems instead of being restricted to the Euclidean distance which behave poorly when we consider clustering a complex nonlinear dataset that resides on a lower dimensional manifold embedded in a high dimensional Euclidean space. Here we discuss an image segmentation method that is proposed in Zhang et al. (2003). This algorithm is developed under the framework of the fuzzy c-means method, it incorporates the spatial information via additional constraints in the objective function and uses the kernel method to represent the similarity measures between pixels.

First of all, we introduce and discuss the kernel method. Let the original dataset be $X = \{\mathbf{x}\}_{j=1}^{N}$. Instead of working directly from the given data representation as in many other algorithms, the kernel methods try to discover more intricate information carried by the dataset by mapping it to some Hilbert space H that is called the feature space. Let the mapping be $\mathbf{f}$, $\mathbf{f} : X \to H$. The mapping $\mathbf{f}$ between the original dataset and H should benefit the comparisons for similarity measures between the original data points. Some direct examples of such mappings are functions of the components of data points (for examples, see (2.20) (2.21) (2.22) (2.23)). The similarity comparison between any pair of the mapped data points $\mathbf{f}(\mathbf{x}_i)$, $\mathbf{f}(\mathbf{x}_j)$ in the Hilbert space H is then calculated in terms of their inner product in the Hilbert space $\langle \mathbf{f}(\mathbf{x}_i), \mathbf{f}(\mathbf{x}_j) \rangle$. Thus the similarity comparison between $\mathbf{x}_i$ and $\mathbf{x}_j$ is defined as the one between $\mathbf{f}(\mathbf{x}_i)$ and $\mathbf{f}(\mathbf{x}_j)$, and finally represented in terms of the inner product in H. In other words, the inner products of the mapped data points are able to define the similarity measures between the original data points. Hence, in order to compare data points in terms of their similarities, it is unnecessary to know the explicit mapping $\mathbf{f}$ from the original dataset to the feature space if we can define the inner products in the feature space.

Now, let us represent the inner products of any pairs of data points in the format of a function, $k : X \times X \to \mathbb{R}$, where $k(\mathbf{x}_i, \mathbf{x}_j) = \langle \mathbf{f}(\mathbf{x}_i), \mathbf{f}(\mathbf{x}_j) \rangle$. The function k is called a kernel. It is important to notice that not all functions can be interpreted as the inner product function of its inputs. The following theorem Jean-Philippe Vert and Scholkopf (2004) give a detailed description of a kernel:

Theorem 3. *If a function k is symmetric and positive semi-definite, then it can be interpreted as the inner product of the mapped data points in the feature space and it is called a positive definite kernel.*

Here, k is symmetric means $k(\mathbf{x}_i, \mathbf{x}_j) = k(\mathbf{x}_j, \mathbf{x}_i)$, for any two data points $\mathbf{x}_i, \mathbf{x}_j$. And k is positive semi-definite means:

$$\sum_{i=1}^{n} \sum_{j=1}^{n} a_i a_j k(\mathbf{x}_i, \mathbf{x}_j) \geq 0$$

for any $n > 0$, any choices of $\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n \in X$ and any choices of real numbers $a_1, a_2, \cdots, a_n \in \mathbb{R}$.

Some simple examples of kernels include

(1). Gaussian radial basis function (RBF) kernel:

$$k_G(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{d(\mathbf{x}, \mathbf{x}')^2}{2\sigma^2}\right) \tag{2.20}$$

(2). Linear kernel:

$$k(\mathbf{x}, \mathbf{x}') = \mathbf{x}^\top \mathbf{x}' \tag{2.21}$$

(3). Polynomial kernels:

$$k_{poly1}(\mathbf{x}, \mathbf{x}') = (\mathbf{x}^\top \mathbf{x}')^d \tag{2.22}$$

or

$$k_{poly2}(\mathbf{x}, \mathbf{x}') = (c + \mathbf{x}^\top \mathbf{x}')^d \tag{2.23}$$

where c is a constant and d is the degree of the polynomials.

More kernels can be generated through operations upon existing kernels or learned from the given datasets (see section 1.6 and 1.7 inJean-Philippe Vert and Scholkopf (2004)).

After the kernel to be used in a clustering algorithm is chosen, every inner product in the feature space can be replaced by the kernel expression. This is the so-called kernel trick. Many clustering algorithms that only use the inner products of data points have been improved with the kernel trick by replacing the inner products of the original data points with the related kernel function values Hoffmann (2007) qiang Zhang and can Chen (2004) Jean-Philippe Vert and Scholkopf (2004). The kernel trick enables the application of clustering algorithm on the mapped dataset $\mathbf{f}(X)$ in the feature space H without knowing the explicit mapping $\mathbf{f}$. One of the most popular kernel methods is Supported Vector Machine (SVM) (see section 1.4 in Jean-Philippe Vert and Scholkopf (2004)).

Now we are ready to discuss a kernelized fuzzy c-means algorithm with spatial constraints qiang Zhang and can Chen (2004).

To cluster a given dataset $X = \{\mathbf{x}_j\}_{j=1}^{N}$ into K clusters, the fuzzy c-means method aims at minimizing the objective function

$$J_m = \sum_{i=1}^{K}\sum_{j=1}^{N} U_{ij}^{m}||\mathbf{x}_j - \mathbf{m}_i||^2$$

Where membership matrix $U = (U_{ij})$ satisfies

$$0 \le U_{ij} \le 1, \quad \forall i,j \quad \text{and} \quad \sum_{i=1}^{K} U_{ij} = 1, \quad \forall j = 1, 2, \cdots, N$$

Since the objective function under minimization only uses the Euclidean distance that can be rewritten as the inner products $\langle \mathbf{x}_j - \mathbf{m}_i, \mathbf{x}_j - \mathbf{m}_i \rangle$ for all i, j, it is suitable to replace the inner product in Euclidean space with the inner product in the feature space H by incorporating the kernel trick. So, we can derive the kernelized fuzzy c-means method by applying fuzzy c-means method in the feature space H. The resulting objective function is:

$$J_m = \sum_{i=1}^{K}\sum_{j=1}^{N} U_{ij}^{m}||\mathbf{f}(\mathbf{x}_j) - \mathbf{f}(\mathbf{m}_i)||^2$$

Notice that

$$\begin{aligned}\|\mathbf{f}(\mathbf{x}_j) - \mathbf{f}(\mathbf{m}_i)\|^2 &= \langle \mathbf{f}(\mathbf{x}_j) - \mathbf{f}(\mathbf{m}_i), \mathbf{f}(\mathbf{x}_j) - \mathbf{f}(\mathbf{m}_i) \rangle \\ &= \langle \mathbf{f}(\mathbf{x}_j), \mathbf{f}(\mathbf{x}_j) \rangle - \langle \mathbf{f}(\mathbf{m}_i), \mathbf{f}(\mathbf{x}_j) \rangle - \langle \mathbf{f}(\mathbf{x}_j), \mathbf{f}(\mathbf{m}_i) \rangle + \langle \mathbf{f}(\mathbf{m}_i), \mathbf{f}(\mathbf{m}_i) \rangle \\ &= k(\mathbf{x}_j, \mathbf{x}_j) + k(\mathbf{m}_i, \mathbf{m}_i) - 2k(\mathbf{x}_j, \mathbf{m}_i)\end{aligned}$$

If one uses the Gaussian kernel as defined in (2.20) from now on, then $k(\mathbf{x}, \mathbf{x}) = 1$ for any $\mathbf{x}$, thus the objective function under minimization can be written as:

$$J_m = 2 \sum_{i=1}^{K} \sum_{j=1}^{N} U_{ij}^m (1 - k(\mathbf{x}_j, \mathbf{m}_i)) \tag{2.24}$$

The constraints used in qiang Zhang and can Chen (2004) are the following:

$$0 \leq U_{ij} \leq 1, \;\; \sum_{i=1}^{K} U_{ij} = 1 \, , \forall j, \;\; \textit{and} \;\; 0 < \sum_{j=1}^{N} U_{ij} < N, \; \forall i \tag{2.25}$$

Following the similar method in minimizing the objective function in fuzzy c-means method, the local minima of J_m for the kernel fuzzy c-means method can be found as:

$$U_{ij} = \frac{(1 - k(\mathbf{x}_j, \mathbf{m}_i))^{-1/(m-1)}}{\sum_{k=1}^{K} (1 - k(\mathbf{x}_j, \mathbf{m}_k))^{-1/(m-1)}} \tag{2.26}$$

and the centroid of the i-th cluster can be found as:

$$\mathbf{m}_i = \frac{\sum_{k=1}^{N} U_{ik}^m k(\mathbf{x}_k, \mathbf{m}_i) \mathbf{x}_k}{\sum_{k=1}^{N} U_{ik}^m k(\mathbf{x}_k, \mathbf{m}_i)} \tag{2.27}$$

Since the performance of segmentation algorithms can be affected if the image is corrupted by noise, spatial information or constraints can be taken into consideration in designing methods that are able to filter noise. On top of such kernelization of the standard fuzzy c-means method, qiang Zhang and can Chen (2004) introduced a spatially constrained term in the objective function to bias the segmentation toward piecewise-homogeneous labeling. The improved objective function requires that for any fixed data point that, for example, is more likely to be clustered in the ith cluster, its neighboring data points should also be more likely to be assigned to the same cluster. To express this idea mathematically, and take into consideration of all the possible assignments of each data point that is weighted by its membership values, the objective function improved with spatial constraints is defined as the following:

$$J_m = \sum_{i=1}^{K} \sum_{j=1}^{N} U_{ij}^m (1 - k(\mathbf{x}_j, \mathbf{m}_i)) + \frac{\alpha}{N_R} \sum_{i=1}^{K} \sum_{j=1}^{N} U_{ij}^m \sum_{r \in N_j} (1 - U_{ir})^m \tag{2.28}$$

With constraints as in (2.25). Here N_j is the set of neighbors in a window around $\mathbf{x}_j$, N_R is the cardinality of N_j. α controls the effect of the spatial penalty term and it is set to be between zero and one.

Similarly, the formulas for calculating membership matrix U and the set of centroids can be derived by using Lagrange multipliers. For a given set of centroids, the membership values can be updated to

$$U_{ij} = \frac{\left(1 - k(\mathbf{x}_j, \mathbf{m}_i) + \alpha \sum_{r \in N_j} (1 - U_{ir})^m / N_R\right)^{-1/(m-1)}}{\sum_{k=1}^{K} \left(1 - k(\mathbf{x}_j, \mathbf{m}_k) + \alpha \sum_{r \in N_j} (1 - U_{kr})^m / N_R\right)^{-1/(m-1)}} \tag{2.29}$$

Since the spatial constrain is not related to any of the centroids $\{\mathbf{m}_i\}_{i=1}^{K}$, then for a given membership matrix U, the calculation for centroids are the same as in (2.27).

Finally, the process to find local minima (as candidates for the global minimization) of the objective function can be summarized in Table 2.3.

The added spatial constraints can effectively reduce the effects of noise in the clustering results. Fig. 2.3 compares the segmentation results on an magnetic resonance image by fuzzy c-means method and the kernel based fuzzy clustering method with spatial constraints (SKFCM) described above. The original image is from the U. S. National Library of Medicine, and it is then corrupted by Gaussian noise with mean 0 and variance 0.0009. Kernel used for the results is the Gaussian radial basis function kernel as in (2.20) with $d(\cdot, \cdot)$ being the Euclidean distance and $2\sigma^2 = 0.15$. The parameter α is set to be 0.15 and the number of neighbors in the window around every data point is $N_R = 8$. That is, all data points inside a 3 by 3 window centered at a fixed data point $\mathbf{x}$ are considered as its neighbors except the data point $\mathbf{x}$ itself. After the initiation by generating a random membership matrix U, the first iteration calculate the set of centroids by using (2.4). After that the calculations follow the procedures described in the algorithm of SKFCM.

Table 2.3: Image segmentation by the kernel-based fuzzy c-means method with spatial constraints (SKFCM)

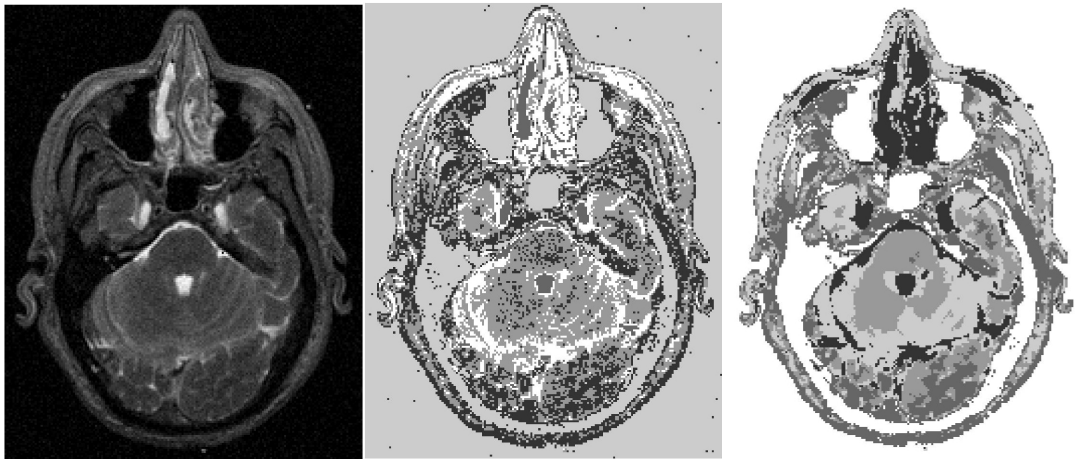
Step 0: Choose a kernel k such that $k(\mathbf{x}, \mathbf{x}')$ is the inner product of the images of $\mathbf{x}$ and $\mathbf{x}'$ in the feature space. In the current discussion, k is chosen to be the Gaussian radial basis function (RBF) kernel:

$$k_G(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{d(\mathbf{x}, \mathbf{x}')^2}{2\sigma^2}\right)$$

Step 1: Initialize the membership matrix U satisfying the constraints as in (2.25).

Step 2: For time $t = 1, 2, \cdots, t_{max}$, do the following iteration:

1. base on the current membership values, compute all centroids according to (2.27).
2. base on the updated centroids, compute all membership values according to (2.29).
3. let the maximum of the updates in membership values be $err^t = \max_{i,j} |U_{ij}^t - U_{ij}^{t-1}|$. If $err^t \leq \epsilon$, stop.



(a) (b) (c)

Figure 2.3: Results of segmentation on a magnetic resonance image corrupted by Gaussian noise with mean 0 and variance 0.0009. (a). The corrupted image. (b). Segmentation result obtained by applying fuzzy c-means method with 5 clusters. (c). Segmentation result obtained by applying SKFCM with 5 clusters. The parameters that used to obtain the results can be found in the text.

2.4.2 The Extension of K-means Method for Overlapping Clustering

Among various methodologies developed to solve clustering problems for large amount of data that are heterogeneous in nature, overlapping clustering arises as one that aims at organizing the dataset into a collection of clusters such that every data point is assigned to at least one cluster and the union of the collection of clusters is the original dataset. Overlapping clustering has some unique properties. It allows different clusters overlapping with each other while crisp clustering does not, and its result only indicates the assignments of each data point to multiple clusters without generating membership values as fuzzy clustering does. These characteristics of overlapping clustering are beneficial in clustering data points which have attributes that can be categorized differently based on multiple perspectives. The applications of overlapping clustering can be recognized in solving various practical problems. For example, information retrieval requires documents to be clustered by domains and each document has multiple domains. Also, in bioinformatics it is likely that the genes to be clustered contribute to multiple metabolic pathwaysCleuziou (2008) Cleuziou (2010).

Different methodologies have been developed through the last four decades for overlapping clustering. Some earlier approaches can be found in Dattola (1968) Jardine and Sibson (1971) Diday (1987). Some more recently research extends existing partition clustering methods to overlapping clustering. For example, the Model-based Overlapping Clustering (MOC) Banerjee et al. (2005a) extends the CEM method proposed in Celleux and Govaert (1992). Also, an extended version of the K-means method, overlapping K-means method (OKM), is proposed in Cleuziou (2008) for overlapping clustering. This section focuses on the description of OKM method proposed in Cleuziou (2008).

For a given dataset $X = \{x_i\}_{i=1}^{N} \subset \mathbb{R}^p$ and a given number of clusters k, OKM aims at minimizing the total distance between data points and their "images" represented by the prototypes of clusters, where the prototypes are the centroids of the clusters. If we denote the clusters as $\{\pi_i\}_{i=1}^{k}$, and the centroids as $\{m_i\}_{i=1}^{k}$, then the goal of OKM is to minimize the

following objective function:

$$J(\{\pi_i\}_{i=1}^k) = \sum_{x_i \in X} \|x_i - \phi(x_i)\|^2 \tag{2.30}$$

where $\phi(x_i)$ is the "image" of x_i denoted by prototypes, i.e. if x_i is assigned to multiple clusters and $A_i = \{m_c \mid x_i \in \pi_c\}$ is the set of centroids of the related clusters, then $\phi(x_i)$ is defined by

$$\phi(x_i) = \frac{\sum_{m_c \in A_i} m_c}{|A_i|} \tag{2.31}$$

and $|A_i|$ is the number of elements in the set A_i.

It is clear to see that the above objective function is a generalization of the one used in the K-means method since the former reduces to the latter when we require that each data point is assigned to only one cluster.

The algorithm that realizes the above minimization of objective function is an iterative procedure that alternates between calculating the centroids of clusters and assigning data points to multiple clusters with the initiation of the algorithm as a random selection of the set of centroids $\{\pi_i^{(0)}\}_{i=1}^k$. The two alternative procedures are described as follows.

When given the set of centroids $\{m_i\}_{i=1}^k$, the assignment of each data point to multiple clusters is not a trivial task because there exists 2^k choices of such multiple assignments for each data point theoretically. Thus a heuristic method is employed such that the assignment of each data point first favors the clusters whose centroids are closest to the data point, and the assignments to clusters are made if the resulting image of the data point is improved in terms of minimizing the objective function. The procedure of assigning a data point x_i to multiple clusters is described in Table 2.4.

Then, the problem of calculating the centroid m_h of a given cluster π_h based on the assignments of data points to multiple clusters and the $k-1$ centroids $\{m_i\}_{i=1}^k \setminus \{m_h\}$ that minimizes $J(\{m_i\}_{i=1}^k)$ can be expressed in a convex minimization problem Cleuziou (2008). And the

Table 2.4: Multiple Assignment of A Data Point

Input:	A data point x_i, the set of centroids $\{m_i\}_{i=1}^k$, and the set of centroids A_i^{old} that corresponds to the assignment of x_i in the previous iteration step.
Output:	A_i, a set of the centroids that defines the multi-assignment for x_i.

1. Let $A_i = \{m^*\}$ such that $m^* = \arg\min_{\{m_i\}_{i=1}^k} \|x_i - m_i\|^2$. Compute $\phi(x_i)$ with the current A_i.
2. Find the nearest centroid: $m' = \arg\min_{\{m_i\}_{i=1}^k \setminus A_i} \|x_i - m_i\|^2$. Compute the image $\phi'(x_i)$ with $A_i \bigcup \{m'\}$.
3. If $\|x_i - \phi'(x_i)\| < \|x_i - \phi(x_i)\|$, let m' be contained in A_i, set $\phi(x_i) = \phi'(x_i)$ and go to step 2; Otherwise compute $\phi^{old}(x_i)$ with A_i^{old}. If $\|x_i - \phi(x_i)\| \leq \|x_i - \phi^{old}(x_i)\|$, output A_i, else output A_i^{old}.

calculation of m_h is as the following:

$$m_h = \frac{1}{\sum_{x_i \in \pi_h} \alpha_i} \sum_{x_i \in \pi_h} \alpha_i m_h^i \tag{2.32}$$

where $\alpha_i = 1/|A_i|^2$ denotes the sharing of x_i among clusters, and m_h^i can be considered as the ideal centroid that matches x_i with its image $\phi(x_i)$:

$$m_h^i = |A_i| \cdot x_i - \sum_{m_c \in A_i \setminus \{m_h\}} m_c$$

Finally, the overall procedure of the OKM algorithm can be summarized in Table 2.5.

Table 2.5: OKM Algorithm

Input: A dataset $X = \{x_i\}_{i=1}^k$, the number of clusters k. Optional maximum number of iterations t_{max}, optional threshold on the objective ϵ.

Output: $\{\pi_i\}_{i=1}^k$ as the final coverage of the dataset.

1. Draw randomly k points in $\mathbb{R}^p$ or X as the initial centroids $\{m_i^{(0)}\}_{i=1}^k$.
2. For each $x_i \in X$ calculate the assignment $A_i^{(0)}$ to derive the initial coverage $\{\pi_i^{(0)}\}_{i=1}^k$.
3. Set $t = 0$.
4. For each cluster $\pi_i^{(t)}$ successively compute the new centroid as in (2.32).
5. For each $x_i \in X$ compute the assignment $A_i^{(t+1)}$ as described above and then derive the coverage $\{\pi_i^{(t+1)}\}_{i=1}^k$.
6. If not converged or $t < t_{max}$ or $J(\{\pi^{(t)}\}) - J(\{\pi^{(t+1)}\}) > \epsilon$, set $t = t + 1$ and go to step 4. Otherwise, stop and output the final clusters $\{\pi_i^{(t+1)}\}_{i=1}^k$.

CHAPTER 3. Spectral Clustering Methods

3.1 Introduction

Spectral clustering has become one of the most important clustering methods over the past decade. The theoretical developments of spectral clustering methods viewed from different aspects have led to successful applications in broad areas of study including color segmentation, protein classification, face recognition, collaborative recommendation and so on Meila and Shi (2000) Cour et al. (2005a) Cour and Shi (2004) Cour et al. (2005b) Enright et al. (2002) Paccanaro et al. (2003). The popularity of spectral clustering are due to but not limited to its profound theoretical supports, convenient implementation, and rich mathematical outreach.

Some standard clustering methods, for example, the K-Means method, usually fail in the case where the datasets reside on nonlinear manifolds that are embedded in higher dimensional spaces. The reason lies in the fact that, these methods like the K-Means method utilize the Euclidean distances in the space occupied by the data points instead of the metric of the underlying manifold. Spectral methods are able to handle complex nonlinear datasets by detecting the intricate data structure, which will be discussed in the following sections. The implementation of the spectral clustering turns out to be easy, and the computational complexity boils down to the calculations of first few eigenvectors of (sparse) matrices Shi and Malik (2000) Luxburg (2007).

Several slightly different versions of spectral clustering methods have been derived in different theoretical background including graph partition Shi and Malik (2000), perturbation theory Stewart and Sun (1990) Luxburg (2007), and stochastic processes on graphs Meila and

Shi (2000) Fouss et al. (2006). The main operator involved in the spectral clustering, the graph Laplacian, can also be seen as a discrete analogy of the Laplace-Beltrami operator under appropriate conditions Singer (2006) Rosenberg (1997) Belkin and Niyogi (2005). Extended studies of spectral clustering give rise to mechanics targeting more specific applications, and efforts have been made to derive variational methods based on spectral clustering Cour and Shi (2004) Bach and Jordan (2004) Cour et al. (2005a).

The outreach of spectral clustering includes its connections to kernel methods and to clustering methods based on random walks. Such connections provide insights into the related mechanisms and allow possible adaptations between these clustering methods Gutman and Xiao (2004) Bie et al. (2006).

Although spectral clustering has advantages as mentioned above, its performance still relies on several factors, for example, how to measure the distances between data points, and how to decide the number of clusters. This chapter is devoted to the review on the spectral clustering methods. The rest of this chapter is arranged as follows: section 2 summarizes the derivation and algorithms of spectral clustering methods, focusing on the connections between spectral clustering and several other related topics including the graph cut problem, random walks on the graph, perturbation theory, and Laplace-Beltrami operator. Section 3 describes some applications of spectral clustering in the fields of image segmentation and shape recognition. Section 4 discusses some techniques that can be used to refine the spectral clustering.

3.2 The Derivations and Algorithms of Spectral Clustering Methods

Spectral clustering methods have been derived under various mathematical settings, and the corresponding algorithms can be classified as unnormalized spectral clustering or normalized spectral clustering. The current section is devoted to the derivations of spectral clustering methods and the descriptions of related algorithms.

3.2.1 Graph Cut and Spectral Clustering

Considering solving a clustering problem on a given dataset $X = \{x_i\}_{i=1}^N$, our goal is to partition the whole dataset into k clusters $C_1, C_2, \cdots, C_k$ such that data points assigned to a same cluster are similar and data points assigned to different clusters are dissimilar. There are many possible choices of measuring the similarities between data points, and such a measure should be decided based on the specific problems. For example, the similarity measure can be defined as the inverse of the Euclidean distances between data points, or be uniform between a data point that is under consideration to its fixed number of nearest data points, etc. A graph $G = (V, E)$ can be constructed to represent data points and show how similar any pairs of data points are. That is, we can take the set of graph nodes V as the dataset X and E as the set of edges between any pairs of nodes in V. Then we can define the weight w_{ij} on the edge which links x_i and x_j as the similarity measure between them. The weight matrix W is an N by N matrix that represents all weights on edges and it is denoted as $W = (w_{ij})_{N\times N}$.

Thus the goal of data clustering on dataset X can be considered as cutting (partitioning) the graph G into k non-overlapping groups such that the weights on inner group edges are high while the weights on intergroup edges are low. To achieve a reasonably good graph cut, measures on the quality of the graph cut should be designed accordingly.

An intuitive measuring method is derived by considering the total similarity between any pairs of different groups. That is, a better graph cut is characterized by a smaller total similarity. If we denote C_i^c as the complement of C_i, then the similarity between a group C_i with any other groups $\{C_j\}_{j\neq i}$ are calculated as

$$W(C_i, C_i^c) = \sum_{u\in C_i, v\in C_i^c} w_{uv} \tag{3.1}$$

Thus, the total similarity between any pairs of different groups can be calculated as the following and it is called the *cut* in graph theoretical language.

$$cut(C_1, C_2, \cdots, C_k) = \frac{1}{2}\sum_{i=1}^{k} W(C_i, C_i^c) \tag{3.2}$$

Here, the factor 1/2 is introduced such that every intergroup edge is considered only once. An algorithm that minimizes the *cut* defined above will result in a partition that allows small similarities between different clusters. For $k = 2$, the problem of finding the minimum *cut* is well studied and Stoer and Wagner (1997) gives an efficient algorithm.

However, as pointed out in Wu and Leahy (2002) and Shi and Malik (2000), the partition generated by minimizing the *cut* tends to have small clusters of isolated data points only because such isolated data points have much lower similarities to any other data points compared to the rest of the dataset. Such clustering results do not provide any further insights on the data structure other than separating out several isolated data points. It is clear that the measure *cut* only takes into account the intergroup similarities and it often fails to serve as a quality measure of the clustering result for general datasets. Two different measures of the clustering quality that are able to circumvent this problem are discussed below. One of them leads to the normalized spectral clustering and the other leads to the unnormalized spectral clustering.

3.2.1.1 Ncut

Shi and Malik proposed a quality measure, called *normalized cut* (*Ncut*) in Shi and Malik (2000), to circumvent the above problem. The idea is to generate "balanced" clusters whose total similarities to the rest of the dataset are small after normalized by their inner-cluster similarities. Let V be the set of nodes, we consider a partition $\{C_1, C_2, \cdots, C_k\}$ of V such that $\bigcup_{i=1}^{k} C_i = V$ and $C_i \bigcap C_j = \emptyset, \ \ \forall i \neq j$. The normalized cut of the partition $\{C_1, C_2, \cdots, C_k\}$ is defined as the following:

$$Ncut(C_1, C_2, \cdots, C_k) = \sum_{i=1}^{k} \frac{cut(C_i, C_i^c)}{assoc(C_i, V)} \tag{3.3}$$

where $assoc(A, B)$ is defined as the total similarity between two groups A and B, and A^c represents the complement of A. That is,

$$assoc(A, B) = \sum_{u \in A, v \in B} w_{uv} \tag{3.4}$$

To verify that the minimization of *Ncut* defined above provides a partition that has minimized intergroup similarities and ensures the results are "balanced" clusters with high innergroup similarities, Shi and Malik (2000) considers the normalized association within groups:

$$Nassoc(C_1, C_2, \cdots, C_k) = \sum_{i=1}^{k} \frac{assoc(C_i, C_i)}{assoc(C_i, V)} \tag{3.5}$$

Thus, a maximized *Nassoc* indicates that the groups in the partition $\{C_1, C_2, \cdots, C_k\}$ have high within group similarities compared to their total similarities to the whole dataset. The relationship between *Nassoc* and *Ncut* is shown by the following:

Proposition 4.

$$Ncut(C_1, C_2, \cdots, C_k) = k - Nassoc(C_1, C_2, \cdots, C_k) \tag{3.6}$$

Proof.

$$\begin{aligned}
Ncut(C_1, C_2, \cdots, C_k) &= \sum_{i=1}^{k} \frac{cut(C_i, C_i^c)}{assoc(C_i, V)} \\
&= \sum_{i=1}^{k} \frac{assoc(C_i, V) - assoc(C_i, C_i)}{assoc(C_i, V)} \\
&= \sum_{i=1}^{k} \left(1 - \frac{assoc(C_i, C_i)}{assoc(C_i, V)}\right) \\
&= k - \sum_{i=1}^{k} \frac{assoc(C_i, C_i)}{assoc(C_i, V)} \\
&= k - Nassoc(C_1, C_2, \cdots, C_k)
\end{aligned}$$

□

Thus, the minimization of *Ncut* also means the maximization of *Nassoc*. Thus it is reasonable to consider *Ncut* an appropriate measure on the quality of a clustering result. Algorithms that minimize *Ncut* are capable of maximizing the within group similarities and minimizing the intergroup similarities simultaneously. Although *Ncut* provides a better quality measure on the clustering results, the exact minimization of *Ncut* is NP-complete Shi and Malik (2000).

To solve the problem, calculations in Shi and Malik (2000) showed that *Ncut* can be rewritten in the form of a Rayleigh quotientGolub and Van Loan (1989) using a type of matrix called graph Laplacian Chung (1997), which will be introduced below. A type of spectral clustering method was designed based on a relaxation of the minimization of this Rayleigh quotient. Shi and Malik (2000) gives details of this derivation, here we follow the procedure provided in Luxburg (2007) for a simplified version to show the relationship between *Ncut* and graph Laplacian.

First, we introduce some notations and discuss some properties of graph Laplacian. More discussions can be found in the book by Chung Chung (1997). Given the weight matrix W, one can define the degree at each node x_i as $d_i = \sum_{j=1}^{N} w_{ij}$. The degree matrix D is defined as

$$D = diag(d_1, d_2, \cdots, d_N) \tag{3.7}$$

The graph Laplacian is defined as

$$L = D - W \tag{3.8}$$

Some properties of the graph Laplacian can be summarized by the following

Proposition 5. *For a given non-negative symmetric weight matrix $W = (w_{ij})_{N\times N}$, the graph Laplacian L is symmetric and positive semi-definite. L has N non-negative eigenvalues $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_N$ and the corresponding eigenvectors $v_1, v_2, \cdots, v_N$ are orthogonal to each other. The smallest eigenvalue is $\lambda_1 = 0$ and the corresponding eigenvector is $\mathbf{1} = (1, 1, \cdots, 1)^T$.*

Proof. Since D is a diagonal matrix, it is clear that $L = D - W$ is symmetric if W is symmetric. Then L has N eigenvalues $\{\lambda_i\}_{i=1}^{N}$ and the corresponding eigenvectors are orthogonal to each other. □

The eigenvectors of L provide information of the number of connected components of the graph G Mohar (1991)Mohar and Juvan (1997).

Proposition 6. *The multiplicity n of the eigenvalue 0 of L equals the number of connected components $A_1, \cdots, A_n$ in the graph G. The eigenspace of eigenvalue 0 is spanned by the indicator vectors $\mathbf{1}_{A_1}, \cdots, \mathbf{1}_{A_n}$.*

Next, for any arbitrary vector $\mathbf{y}$ in $\mathbb{R}^N$,

$$\begin{aligned} 2\mathbf{y}^T L\mathbf{y} &= 2(\mathbf{y}^T D\mathbf{y} - \mathbf{y}^T W\mathbf{y}) \\ &= \sum_{i=1}^{N} d_i y_i^2 - 2\sum_{i,j=1}^{N} w_{ij} y_i y_j + \sum_{j=1}^{N} d_j y_j^2 \\ &= \sum_{i=1}^{N}\sum_{j=1}^{N} w_{ij} y_i^2 - 2\sum_{i,j=1}^{N} w_{ij} y_i y_j + \sum_{j=1}^{N}\sum_{i=1}^{N} w_{ij} y_j^2 \\ &= \sum_{i,j=1}^{N} w_{ij}(y_i - y_j)^2 \end{aligned}$$

Since W is non-negative, $\mathbf{y}^T L\mathbf{y} \geq 0$ for any arbitrary vector $\mathbf{y}$ in $\mathbb{R}^N$. That is, L is positive semi-definite. Thus, all eigenvalues of L are non-negative and it is easy to see that the smallest eigenvalue is 0 corresponding to an eigenvector $\mathbf{1} = (1, 1, \cdots, 1)^T$.

We now consider the case of bipartition of the graph ($k = 2$). The analogical conclusions in cases where $k \geq 2$ can be derived similarly Luxburg (2007).

Now consider the bipartition (the case where k=2) of a graph G using *Ncut* as the quality measure (the cases where $k \geq 3$ will be discussed later). For a bipartition $\{C_1, C_2\}$ of the set of nodes V ($C_1 \cup C_2 = V$ and $C_1 \cap C_2 = \emptyset$) one can define the indicator vector as $\mathbf{f} = (f_1, f_2, \cdots, f_N)^T$ such that $\mathbf{f}$ carries the bipartition information. That is, $f_j = a$ for $x_j \in C_1$ and $f_j = b$ for $x_j \in C_2$, where a, b are two different constants. If one chooses to define $\mathbf{f}$ as

$$f_i = \begin{cases} \sqrt{\frac{vol(C_2)}{vol(C_1)}} & \text{if } x_i \in C_1 \\ -\sqrt{\frac{vol(C_1)}{vol(C_2)}} & \text{if } x_i \in C_2 \end{cases} \tag{3.9}$$

where we adopt the notation $vol(A) = assoc(A, V)$. Then the following calculation reveals the

relationship between graph Laplacian and *Ncut*, here we use $i \in C_j$ to represent $x_i \in C_j$, $\forall i, j$:

$$
\begin{aligned}
2\mathbf{f}^T L\mathbf{f} &= \sum_{i,j=1}^{N} w_{ij}(f_i - f_j)^2 \\
&= \sum_{i\in C_1, j\in C_2} w_{ij}\left(\sqrt{\frac{vol(C_2)}{vol(C_1)}} + \sqrt{\frac{vol(C_1)}{vol(C_2)}}\right)^2 + \sum_{i\in C_2, j\in C_1} w_{ij}\left(-\sqrt{\frac{vol(C_1)}{vol(C_2)}} - \sqrt{\frac{vol(C_2)}{vol(C_1)}}\right)^2 \\
&= \sum_{i\in C_1, j\in C_2} w_{ij}\left(\frac{vol(C_2)}{vol(C_1)} + \frac{vol(C_1)}{vol(C_2)} + 2\right) + \sum_{i\in C_2, j\in C_1} w_{ij}\left(\frac{vol(C_1)}{vol(C_2)} + \frac{vol(C_2)}{vol(C_1)} + 2\right) \\
&= 2\left(\frac{vol(C_1)}{vol(C_2)} + \frac{vol(C_2)}{vol(C_1)} + 2\right) cut(C_1, C_2) \\
&= 2\left(\frac{vol(C_1) + vol(C_2)}{vol(C_2)} + \frac{vol(C_1) + vol(C_2)}{vol(C_1)}\right) cut(C_1, C_2) \\
&= 2vol(V)Ncut(C_1, C_2)
\end{aligned}
$$

That is ,

$$\mathbf{f}^T L\mathbf{f} = vol(V)Ncut(C_1, C_2) \tag{3.10}$$

Also one can check that $(D\mathbf{f})^T\mathbf{1} = 0$ and $\mathbf{f}^T D\mathbf{f} = vol(V)$. Then by allowing $\mathbf{f}$ to be an arbitrary vector in $\mathbb{R}^N$, one can derive a relaxation of the minimization of *Ncut*.

$$\min_{\mathbf{f}\in\mathbb{R}^N} \mathbf{f}^T Lf \quad \text{subject to } \; D\mathbf{f} \perp \mathbf{1}, \;\; \mathbf{f}^T D\mathbf{f} = vol(V) \tag{3.11}$$

Now we can introduce the following substitution $\mathbf{g} = D^{1/2}\mathbf{f}$ and the above problem can be rewritten as

$$\mathbf{g}^T D^{-1/2} L D^{-1/2}\mathbf{g}$$

$$\text{subject to } \; \mathbf{g} \perp D^{1/2}\mathbf{1}, \;\; ||\mathbf{g}||^2 = vol(V) \tag{3.12}$$

$D^{-1/2}LD^{-1/2}$ is symmetric and it is a version of **normalized graph Laplacian**. It can be denoted as

$$L_{sym} = D^{-1/2}LD^{-1/2} \tag{3.13}$$

A similar calculation reveals that for arbitrary $\mathbf{y} \in \mathbb{R}^N$,

$$\mathbf{y}^T L_{sym}\mathbf{y} = \sum_{i,j=1}^{N} w_{ij}\left(\frac{y_i}{\sqrt{d_i}} - \frac{y_j}{\sqrt{d_j}}\right)^2$$

Thus, L_{sym} is positive semi-definite. Use the fact that it is symmetric, we can conclude that L_{sym} has N non-negative eigenvalues $\{\lambda_i\}_{i=1}^N$ with eigenvectors $\{\mathbf{v_i}\}_{i=1}^N$ orthogonal to each other. Also, the smallest eigenvalue is $\lambda_1 = 0$ with at least one corresponding eigenvector $\mathbf{v_1} = D^{1/2}\mathbf{1}$.

Similar to Proposition 6, L_{sym} has properties described as follows:

Proposition 7. *The multiplicity n of the eigenvalue 0 of L_{sym} equals the number of connected components $A_1, A_2, \cdots, A_n$ in the graph G. The eigenspace of eigenvalue 0 is spanned by the indicator vectors $D^{1/2}\mathbf{1}_{A_1}, D^{1/2}\mathbf{1}_{A_2}, \cdots, D^{1/2}\mathbf{1}_{A_n}$.*

Considering the above properties of L_{sym}, the minimization in (3.12) can be transformed to the standard form of the Rayleigh quotientGolub and Van Loan (1989) in finding $\mathbf{g_1}$ such that:

$$\mathbf{g_1} = \arg\min_{\mathbf{g}^T\mathbf{g_0}=0} \frac{\mathbf{g}^T D^{-1/2} L D^{-1/2}\mathbf{g}}{\mathbf{g}^T\mathbf{g}} \tag{3.14}$$

where $\mathbf{g_0} = D^{1/2}\mathbf{1}$.

Notice that in the current setting we assume that similarity measures of any pairs of nodes are nonzero, thus the graph G is complete. By Proposition 7, $D^{1/2}\mathbf{1}$ is the only eigenvector that corresponds to eigenvalue 0. Thus the solution is parallel to the eigenvector of L_{sym} that corresponds to the smallest positive eigenvalue. Also, the above is equivalent to finding f_1 such that:

$$\mathbf{f_1} = \arg\min_{\mathbf{f}^T D\mathbf{1}} \frac{\mathbf{f}^T(D-W)\mathbf{f}}{\mathbf{f}^T D\mathbf{f}} \tag{3.15}$$

Then this problem can be solved by finding the eigenvector $\mathbf{f}$ corresponds to the smallest positive eigenvalue of the generalized eigenvalue problem:

$$L\mathbf{f} = \lambda D\mathbf{f} \tag{3.16}$$

The solution $\mathbf{f_1}$ serves as an indicator vector for the bipartition of V. As mentioned before, in the cases where the number of clusters $k > 2$ the above procedure holds due to similar arguments. Then the partition of components of $\mathbf{f_1}$ corresponds to the optimized partition of V.

Similar to the procedure of finding $\mathbf{f_1}$, the next eigenvector of (3.16) can be considered as a second best indicator vector for the partition. Assume one takes into account the m eigenvectors of (3.16) corresponding to the smallest m positive eigenvalues to decide the final partition, the algorithm can be summarized as follows. Since this procedure relies on spectral graph theory, the resulting algorithm is classified as spectral clustering method.

Normalized spectral clustering introduced in Shi and Malik (2000):

1. For a given complete graph $G = (V, E)$ with weight matrix W, compute the m eigenvectors corresponding to the smallest m positive eigenvalues of the generalized eigenproblem $L\mathbf{f} = \lambda D\mathbf{f}$, denote these eigenvectors as $\mathbf{f_1}, \cdots, \mathbf{f_m}$.

2. Denote the new representation of x_i by $y_i = (\mathbf{f_1}(i), \mathbf{f_2}(i), \cdots, \mathbf{f_m}(i))$.

3. Apply a standard data clustering method, eg. K-Means method, to partition the new dataset $(y_i)_{i=1}^N$ into k groups.

4. For $i \in \{1, 2, \cdots, N\}$, assign x_i to the j-th cluster C_j if y_i is assigned to the j-th cluster in the previous step.

Notice that if G has more than one connected components and if it is reasonable to partition G base on these components, then by Proposition 7, all the indicator vectors of the clusters should be found as all the eigenvectors that correspond to the eigenvalue 0 of (3.16). And the step 1 in the above algorithm should be changed accordingly.

3.2.1.2 RatioCut

Another type of measure called *RatioCut* is proposed in Hagen (1992) based on a similar idea of generating "balanced" clusters:

$$RatioCut(C_1, C_2, \cdots, C_k) = \frac{1}{2}\sum_{i=1}^{k} \frac{cut(C_i, C_i^c)}{|C_i|} \tag{3.17}$$

where $|A|$ is the number of nodes in the set A. The minimization of *RatioCut* is also NP-complete.

Take the case where $k = 2$ as an example, and we still take G as a complete graph provided with similarity measures on each pair of nodes. Similar to the procedures taken for *Ncut*, the *RatioCut* can be rewritten using the graph Laplacian L conveniently. Define vector $\mathbf{f} = (f_1, f_2, \cdots, f_N)^T$ as

$$f_i = \begin{cases} \sqrt{|C_2|/|C_1|} & \text{if } x_i \in C_1 \\ -\sqrt{|C_1|/|C_2|} & \text{if } x_i \in C_2 \end{cases} \tag{3.18}$$

Then by a similar calculation as before,

$$\mathbf{f}^T L\mathbf{f} = |V| \cdot RatioCut(C_1, C_2) \tag{3.19}$$

and it is easy to see $\sum_{i=1}^{N} f_i = 0$, $\mathbf{f}^T\mathbf{f} = N$. Then this NP-complete minimization problem can be summarized as

$$\min_{C_1 \cap C_2 = \emptyset, C_1 \cup C_2 = V} \mathbf{f}^T L\mathbf{f}$$

$$\text{subject to } \mathbf{f} \perp \mathbf{1}, \quad f_i \text{ defined as in (3.18)}, \ ||\mathbf{f}|| = \sqrt{N} \tag{3.20}$$

The relaxation on the condition (3.18) by allowing f_i to take arbitrary value in $\mathbb{R}$ transforms the above problem to

$$\min_{\mathbf{f} \in \mathbb{R}^N} \mathbf{f}^T L\mathbf{f}$$

$$\text{subject to } \mathbf{f} \perp \mathbf{1}, \quad ||\mathbf{f}|| = \sqrt{N} \tag{3.21}$$

By Rayleigh-Ritz theorem, the solution is parallel to the eigenvector that corresponds to the smallest positive eigenvalue of L. The partition on the components of $\mathbf{f}$ corresponds to the optimized partition of V by considering the quality measure *RatioCut*. In the cases where $k > 2$, similar arguments lead to the same conclusionLuxburg (2007). The clustering procedure can be summarized as the following unnormalized spectral clustering algorithm.

Unnormalized Spectral Clustering:

1. Given a complete graph G with the similarity measures between each pair of nodes, construct the weight matirx W. Compute the m eigenvectors $\mathbf{v_1}, \mathbf{v_2}, \cdots, \mathbf{v_m}$ of L that correspond to the m smallest positive eigenvalues.

2. Denote the new representation of x_i by $y_i = (\mathbf{v_1}(i), \mathbf{v_2}(i), \cdots, \mathbf{v_m}(i))$.

3. Apply a standard data clustering method, eg. K-Means method, to partition the new dataset $(y_i)_{i=1}^N$ into k clusters.

4. Assign x_i to the j-th cluster C_j if y_i is assigned to the j-th cluster in the previous step.

The treatment for the cases where graph G has more than one connected components is similar to that mentioned in the previous section.

The above relaxation procedures on minimization of *Ncut* or *RatioCut* lead to efficient algorithms that only involve solving standard linear algebra problems. However, the qualities of such clustering methods should be evaluate carefully because the clustering results are affected by the relaxations. Some of such evaluations are described in Luxburg (2007), Kannan et al. (2000) and Spielman (1996). The relaxation on the minimization problems of *Ncut* or *RatioCut* is not unique and results have shown that the approximation to the optimized partition using a balanced measure of graph cuts may be NP hard itselfBui and Jones (1992).

3.2.2 Random Walks and Spectral Clustering

Random walks on a graph can provide another justification for spectral clustering. The graph $G = (V, E)$ is still constructed by taking the dataset X as the set of nodes and E as the set of edges between nodes. Then the similarities measured between any pair of data points provide a natural transition matrix P for the random walk on G:

$$P = D^{-1}W \tag{3.22}$$

where D is the degree matrix defined in (3.7). Every row sum of P is 1 and the transition probability p_{ij} is proportional to the similarity between x_i and x_j. That is, the probability of the

random walk jumping to x_j starting from x_i is proportional to the similarity between them. If G is connected, then there exists a unique stationary distribution $\pi = (\pi_1, \pi_2, \cdots, \pi_N)^T$ with $\pi_i = d_i/vol(V)$. We can consider the random walk $\{X_t\}_{t\in\mathbb{N}}$, starting from X_0 in the stationary distribution π. Then a good graph partition $\{C_1, C_2, \cdots, C_k\}$ should have the properties that the probabilities of taking random walks in between of different groups should be small compared to those within a same group. This can be seen by the following calculation where we take the simple case $k = 2$:

$$\begin{aligned}
P(X_1 \in C_2 \mid X_0 \in C_1) &= \frac{P(X_1 \in C_2, X_0 \in C_1)}{P(X_0 \in C_1)} \\
&= (P(X_0 \in C_1, X_1 \in C_2)) \left(\frac{vol(C_1)}{vol(V)}\right)^{-1} \\
&= \sum_{i\in C_1, j\in C_2} P(X_0 = i, X_1 = j) \left(\frac{vol(V)}{vol(C_1)}\right) \\
&= \sum_{i\in C_1, j\in C_2} \pi_i p_{ij} \left(\frac{vol(V)}{vol(C_1)}\right) \\
&= \sum_{i\in C_1, j\in C_2} \frac{d_i}{vol(V)} \frac{w_{ij}}{d_i} \left(\frac{vol(V)}{vol(C_1)}\right) \\
&= \frac{\sum_{i\in C_1, j\in C_2} w_{ij}}{vol(V)} \left(\frac{vol(V)}{vol(C_1)}\right) \\
&= \frac{\sum_{i\in C_1, j\in C_2} w_{ij}}{vol(C_1)}
\end{aligned}$$

Similarly,

$$P(X_1 \in C_1 \mid X_0 \in C_2) = \frac{\sum_{i\in C_2, j\in C_1} w_{ij}}{vol(C_2)}$$

Then by the definition of $Ncut$ in (3.3)

$$P(X_1 \in C_1 \mid X_0 \in C_2) + P(X_1 \in C_2 \mid X_0 \in C_1) = Ncut(C_1, C_2) \tag{3.23}$$

Random walk on the graph can be connected to another form of normalized graph Laplacian $D^{-1}L$. Denote

$$L_{rw} = D^{-1}L = I - D^{-1}W \tag{3.24}$$

Then it is easy to see $L_{rw} = I - P$. Thus, $(\lambda, \mathbf{v})$ is an eigenpair of P if and only if $(1 - \lambda, \mathbf{v})$ is an eigenpair of L_{rw}. Because many properties of the random walk on the graph rely on the transition matrix P and its eigenvalues and eigenvectors, the connection between the random

walk and L_{rw} becomes obvious through their connection of eigenspaces. Some properties of L_{rw} and its connections with L and L_{sym} can be summarized in the following:

Proposition 8. *$(\lambda, \mathbf{v})$ is an eigenpair of L_{rw} if and only if $(\lambda, D^{1/2}\mathbf{v})$ is an eigenpair of L_{sym}. It is also equivalent to that $(\lambda, \mathbf{v})$ solves the generalized eigenproblem $L\mathbf{v} = \lambda D\mathbf{v}$. Especially, $(0, \mathbf{1})$ is an eigenpair of L_{rw}. L_{rw} is positive semi-definite and has N non-negative real-valued eigenvalues $0 = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_N$.*

Proof. The last part of the above statements can be seen by the following calculation

$$2\mathbf{y}^T L_{rw}\mathbf{y} = \sum_{i,j=1}^{N} w_{ij}\left(\frac{y_i}{\sqrt{d_i}} - \frac{y_j}{\sqrt{d_i}}\right)^2$$

for an arbitrary $\mathbf{y} \in \mathbb{R}^N$. Thus, L_{rw} is positive semi-definite. The rest part of the statements are easy to verify. □

Due to the above discussion, random walks on the graph provides a natural justification of the normalized spectral clustering methodsMeila and Shi (2000) Luxburg (2007). It also provides insights in designing other types of clustering methods Fouss et al. (2006) which will be discussed later.

3.2.3 Perturbation Theory and Spectral Clustering

For a given well separated dataset X that consists of connected components $\{A_1, A_2, \cdots, A_k\}$, the ideal case is referring to when the given similarity measures between any pair of nodes correctly carry the information of the different connected components of X. That is, the weight matrix W and the corresponding graph Laplacian L is block diagonal after a permutation on the nodes such that nodes in a same component are arranged together. By the Proposition 6, 7 and 8, the indicator vectors $\{\mathbf{1}_{A_1}, \mathbf{1}_{A_2}, \cdots, \mathbf{1}_{A_k}\}$ of the connected components are eigenvectors that correspond to the eigenvalue 0 of L and L_{rw}. However, in general the graph Laplacian $\tilde{L}$ is usually not block diagonal. It can be considered as in the form of the ideal case with a perturbation $\tilde{L} = L + H$, where L is the graph Laplacian in the ideal case and H is a perturbation. Then the first k eigenvectors $\{\tilde{\mathbf{v}}_1, \tilde{\mathbf{v}}_2, \cdots, \tilde{\mathbf{v}}_k\}$ of the perturbed $\tilde{L}$ can not be found as the k indicator vectors of clusters anymore. The clustering base on $\{\tilde{\mathbf{v}}_i\}_{i=1}^k$ can lead to reasonable

partition of X if $\{\tilde{\mathbf{v}}_i\}_{i=1}^k$ can be found as in the form of $\{\mathbf{1}_{A_i}\}_{i=1}^k$ with small perturbations. The comparison of two sets of eigenvectors are actually the comparison of the two subspaces spanned by them. Thus, good clustering results can be derived if the distance between the subspaces spanned by $\{\mathbf{1}_{A_i}\}_{i=1}^k$ and $\{\tilde{\mathbf{v}}_i\}_{i=1}^k$ is small.

Perturbation theory provides results that can be used to measure a distance of subspaces of the eigenspaces of L and $\tilde{L}$. In perturbation theory, a distance between two p dimensional subspaces $\mathcal{V}_1$ and $\mathcal{V}_2$ in $\mathbb{R}^d$ is measured through the two matrices V_1, V_2 whose columns form orthogonal systems for $\mathcal{V}_1$ and $\mathcal{V}_2$ respectively. The cosines $\cos\Theta_i$ of the "principle angle" Θ_i Stewart and Sun (1990) are the singular values of $V_1^T V_2$. The following theorem by Davis-Kahan Stewart and Sun (1990) gives an upper bound for the distance between V_1, V_2.

Theorem 9. *[**Davis-Kahan**] Let A, $H \in \mathbb{R}^{n\times n}$ be symmetric matrices, and let $||\cdot||$ be the Frobenius norm or the two-norm for matrices, respectively. Consider $\tilde{A} = A + H$ as a perturbed version of A. Let $S_1 \subset \mathbb{R}$ be an interval. Denote by $\sigma_{S_1}(A)$ the set of eigenvalues of A which are contained in S_1, and by V_1 the eigenspace corresponding to all those eigenvalues. Denote by $\sigma_{S_1}(\tilde{A})$ and $\tilde{V}_1$ the analogous quantities for $\tilde{A}$. Define the distance between S_1 and the spectrum of A outside of S_1 as*

$$\delta = \min\{|\lambda - s| \ ; \ \lambda \text{ eigenvalue of } A, \ \lambda \notin S_1, \ s \in S_1\}$$

Then the distance $d(V_1, \tilde{V}_1) := ||\sin\Theta(V_1, \tilde{V}_1)||$ between the two subspaces $V_1, \tilde{V}_1$ is bounded by

$$d(V_1, \tilde{V}_1) \leq \frac{||H||}{\delta}$$

This theorem provides results under the conditions that A is the matrix whose first few eigenvectors are indicating vectors of the connected components of the given graph (this is called "the ideal case" as mentioned before), and the perturbation is small (i.e. if $||H||$ is small). If it is possible to find an interval S_1 such that the first k eigenvalues of L and $\tilde{L}$ are all contained in S_1 with the eigengap $\delta = |\lambda_{k+1} - \lambda_k|$ big enough, then the above theorem indicates that the distance between the subspaces spanned by $\{\tilde{\mathbf{v}}_i\}_{i=1}^k$ and $\{\mathbf{1}_{A_i}\}_{i=1}^k$ are bounded by $\frac{||H||}{\delta}$ from above. If this distance $\frac{||H||}{\delta}$ is small, then for every i, $\tilde{\mathbf{v}}_i$ is almost constant on

each connected component of the graph because it can be almost considered as the result of some linear combination of $\{\mathbf{1}_{A_i}\}_{i=1}^{k}$. Thus, by applying a standard clustering method on the rows of $\tilde{V} = [\tilde{\mathbf{v}}_1, \tilde{\mathbf{v}}_2, \cdots, \tilde{\mathbf{v}}_k]$ one can generate reasonable clustering results.

This upper bound given in the above theorem provides some estimation on the quality of the partition of X by clustering the first k eigenvectors $\{\tilde{\mathbf{v}}_i\}_{i=1}^{k}$. Also it provides a method for determining how many first eigenvectors $\{\tilde{\mathbf{v}}_1, \tilde{\mathbf{v}}_2, \cdots, \tilde{\mathbf{v}}_{k_0}\}$ could be used to possibly achieve good clustering results when the number of connected components is not clear. That is, to achieve possibly good clustering results, the number of nontrivial eigenvectors k_0 can be chosen such that λ_{k_0} has the biggest eigengap with its next eigenvalue λ_{k_0+1}:

$$k_0 = \arg \max_{k=1,2,\cdots,N} \{|\lambda_k - \lambda_{k+1}|\}$$

Notice that, as mentioned in Proposition 7, in the ideal case the first k eigenvectors of L_{sym} are $V = [D^{1/2}\mathbf{1}_{A_1}, D^{1/2}\mathbf{1}_{A_2}, \cdots, D^{1/2}\mathbf{1}_{A_k}]$. Then the differences in the degrees of nodes will affect the clustering results. Another version of normalized spectral clustering proposed by Ng et al. in Ng et al. (2001) circumvents this problem by normalizing each row of $\tilde{V} = [\tilde{\mathbf{v}}_1, \tilde{\mathbf{v}}_2, \cdots, \tilde{\mathbf{v}}_k]$. It can be summarized as the following: (All the matrices and eigenvectors mentioned below need not to correspond to the ideal case, we simply omitted the ˜ symbol in the following describtion.)

Normalized spectral clustering introduced in Ng et al. (2001):

1. For a given dataset X and similarities between pairs of nodes, construct the weight matrix W and calculate L_{sym}. Choose an appropriate number m as the number of eigenvectors for the following clustering. Compute the first m eigenvectors $\mathbf{v}_1, \cdots, \mathbf{v}_m$ of L_{sym}. Form the matrix $V = [\mathbf{v}_1, \cdots, \mathbf{v}_m]$ and construct the matrix T by normalizing each row of V to norm 1. That is, set
$$t_{ij} = \frac{v_{ij}}{(\sum_{k=1}^{N} v_{ik}^2)^{1/2}}$$
2. Denote the new representation of x_i by y_i, where y_i is the i-th row of T.

3. Apply a standard data clustering method, eg. K-Means method, to partition the new dataset $(y_i)_{i=1}^N$ into k clusters.

4. Assign x_i to the j-th cluster C_j if y_i is assigned to the j-th cluster in the previous step.

Although the above algorithm tries to circumvent the problem of difference in degrees, there are cases where it still cannot help to derive the correct clusteringLuxburg (2007).

As a summary, perturbation theory supports the spectral clustering methods that utilize the eigenvectors of L and L_{rw}. The normalization procedure in the above algorithm based on L_{sym} is also justified. However, the above algorithm should be applied with care if there are nodes in the graph that have particularly small degrees. The reason is, such nodes will become indistinguishable after the row normalization step in the above algorithm (for details, seeLuxburg (2007)). On the other hand, as mentioned in Luxburg (2007) again, such nodes may be preprocessed due to their low degrees (for example, classified in to a group labeled as outliers) before applying spectral algorithms.

3.2.4 The Graph Laplacian and Laplace-Beltrami Operator

3.2.4.1 Embedded Manifolds and the Graph Laplacian

In previous sections, similarity measures between data points are considered as one of the most fundamental information for the construction of the weight matrix. It has also been implicitly assumed that the chosen similarity measures utilized in clustering algorithms are able to correctly reveal the geometric properties of the data structures. However, this assumption should not be taken for granted, especially when the datasets collected from experiments or surveys reside on manifolds that are embedded in high dimensional spaces. To be more specific, if a dataset X in $\mathbb{R}^D$ actually resides on an m-dimensional manifold $\mathcal{M}$ ($m < D$) equipped with metric $d_{\mathcal{M}}$, then it is reasonable to determine similarity measures using the metric $d_{\mathcal{M}}$ on $\mathcal{M}$ rather than the Euclidean distance defined in $\mathbb{R}^D$. An example of this can be found in the face recognition problem, which can be formulated as a problem of clustering digital facial images taken under various circumstances into groups corresponding to different individuals. If the

digital images are of n by n dimension, then the set of images are in $\mathbb{R}^{n^2}$, which is usually a high dimensional space. However, it has been shown that facial images reside on lower dimensional nonlinear submanifolds Chang et al. (2003a) Lee et al. (2003a) Roweis and Saul (2000a) Roweis et al. (2002a) Seung and Lee (2000) Shashua et al. (2002) Tenenbaum et al. (2000). Thus, it would be efficient and accurate to measure similarities between data points using the metric defined on the underlying manifold. In applications, methods developed for solving face recognition problems that attempt to detect the embedded manifolds and represent the images with new sets of basis in lower dimensional spaces have been proved effective Moghaddam (2002) del Solar and Navarrete (2005) Turk and Pentland (1991a) Belhumeur et al. (1997) He et al. (2005).

The procedure of discovering the underlying manifold $\mathcal{M}$ and its metric $d_{\mathcal{M}}$ (if they exist) of a given dataset $X = \{\mathbf{x}_1, \cdots, \mathbf{x}_N\}$ embedded in $\mathbb{R}^D$ can be considered as the following. For each i in $\{1, 2, \cdots, N\}$, one aims at finding a representation $\mathbf{z}_i$ for $\mathbf{x}_i$ in a space $\mathbb{R}^L$ while trying to preserve the Euclidean metric on $\{\mathbf{z}_1, \cdots, \mathbf{z}_N\}$ in $\mathbb{R}^L$ as the metric d_M on X Bronstein et al. (2008). In other words, it is to find an isometry $f : X \to Z \subset \mathbb{R}^L$, such that

$$d_{\mathcal{M}}(\mathbf{x}_i, \mathbf{x}_j) = d_{\mathbb{R}^L}(\mathbf{z}_i, \mathbf{z}_j)$$

where $d_{\mathbb{R}^L}(\cdot, \cdot)$ is the Euclidean metric in $\mathbb{R}^L$. Nash's embedding theorem Nash (1954) guarantees that any n-dimensional manifold $\mathcal{M}$ can be isometrically embedded in a $2n+1$-dimensional Euclidean space through a C^1 isometry. However, this type of isometric embedding is not unique. The reason is, if we assume $\mathcal{M}$ is mapped isometrically as $\mathcal{N}$ in $\mathbb{R}^{(2n+1)}$, then any isometry $g : \mathbb{R}^{(2n+1)} \to \mathbb{R}^{2n+1}$ that is not identity operator can generate another isometric embedding $g(\mathcal{N})$ of $\mathcal{M}$. In addition, such isometric embedding results in a new dataset of a higher dimension, which is not beneficial for problems like face recognition.

A more practical approach that "approximates" the above isometric embedding by preserving *local* metric properties of the manifold can be described as follows.

For a given dataset $X = \{\mathbf{x}_1, \cdots, \mathbf{x}_N\}$ with metric d_X, we aim at mapping X to its new

representation $\{\mathbf{z}_1, \cdots, \mathbf{z}_N\} \subset \mathbb{R}^p$ such that the neighboring points in the original dataset are still neighbors in the new dataset. More formally, we can construct a graph by taking the set of vertices as X and choose a definition for neighbors (for example, define neighbors as a fixed number of nearest data points, or the data points within a fixed Euclidean distance). Then we denote W as the weight matrix and w_{ij} as the weight on the edge between $\mathbf{x}_i$ and $\mathbf{x}_j$. Assign nonnegative weights on edges that link between neighbors and leave the weights on other edges 0 . For example, a simple choice of weight matrix is the following:

$$w_{ij} = \begin{cases} 1 & \text{if } \mathbf{x}_i \text{ and } \mathbf{x}_j \text{ are neighbors} \\ 0 & \text{otherwise} \end{cases} \tag{3.25}$$

Below is another choice of weight matrix that will soon be shown suggested by the heat kernelBelkin and Niyogi (2003):

$$w_{ij} = \begin{cases} \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{4t}\right) & \text{if } \mathbf{x}_i \text{ and } \mathbf{x}_j \text{ are neighbors} \\ 0 & \text{otherwise} \end{cases} \tag{3.26}$$

If $\mathbf{x}_i$ has a neighbor $\mathbf{x}_j$ and $d_X(\mathbf{x}_i, \mathbf{x}_j)$ is small, we require that the distances between $\mathbf{z}_i$ and $\mathbf{z}_j$ is also small in $\mathbb{R}^p$. Combining such requirements at each data point, we can formulate the following optimization problem to find the new representation of the original dataset:

$$Z^* = \arg\min_{Z \in \mathbb{R}^{N \times p}} \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} w_{ij} d^2(\mathbf{z}_i, \mathbf{z}_j) \tag{3.27}$$

where the new representation is denoted as $Z = [\mathbf{z}_1, \mathbf{z}_2, \cdots, \mathbf{z}_N]^T$, and $d(\cdot, \cdot)$ is the Euclidean distance.

To avoid the case where every data point in X is embedded as a same point in $\mathbb{R}^p$, one can pose a constraint as $Z^T Z = I$ (I is the identity matrix). Next, we can transform the above optimization problem using the graph Laplacian. That is, if the degree matrix and graph Laplacian are defined as in (3.7) and (3.8), then the above optimization problem becomes

$$Z^* = \arg\min_{Z \in \mathbb{R}^{N \times p}} trace(Z^T L Z) \tag{3.28}$$

with constraint $Z^T Z = I$.

For a fixed p, Z^* in $\mathbb{R}^{N\times p}$ can be proved as $[\mathbf{y}_1, \cdots, \mathbf{y}_p]$, where $\mathbf{y}_i$ is the eigenvector of L corresponding with the ith smallest positive eigenvalueBronstein et al. (2008).

3.2.4.2 The Graph Laplacian and Laplace-Beltrami Operator

The Laplace-Beltrami operator is a continuous analogy of graph LaplacianBelkin and Niyogi (2003) Bronstein et al. (2008). Here we follow Belkin and Niyogi (2003) and Bronstein et al. (2008) to describe this relationship and induce a choice of weight matrix suggested by heat kernel.

For the current discussion, we assume that the m-dimensional manifold $\mathcal{M}$ is smooth, compact and is isometrically embedded in a higher dimensional space $\mathbb{R}^D$ and its metric $d_{\mathcal{M}}$ is induced by the standard Riemannian structure in $\mathbb{R}^D$. We aim at finding a map $f : \mathcal{M} \to \mathbb{R}$ such that, roughly speaking, points that are close to each other are mapped on the real line and their new representations are also close together. We also require that f is smooth enough, for example, f is twice differentiable.

Let $\mathbf{x}$ and $\mathbf{y}$ are two neighboring points on $\mathcal{M}$, we have the following Proposition as an estimation of the distance between their new representations after the mapping:

Proposition 10. *If* $\mathbf{x}$ *and* $\mathbf{y}$ *are neighboring points on* $\mathcal{M}$, $f : \mathcal{M} \to \mathbb{R}$ *is twice differentiable, then*

$$|f(\mathbf{x}) - f(\mathbf{y})| \leq ||\nabla f(\mathbf{x})|| \cdot d_{\mathcal{M}}(\mathbf{x}, \mathbf{y}) + o(d_{\mathcal{M}}(\mathbf{x}, \mathbf{y})) \tag{3.29}$$

Proof. Let the tangent space of the manifold $\mathcal{M}$ at point $\mathbf{x}$ be $T\mathcal{M}_x$. For any vector $\mathbf{v}$ in $T\mathcal{M}_x$, we define the *differential* $df(\mathbf{v})$ as $df(\mathbf{v}) = \langle \nabla f, \mathbf{v} \rangle_{\mathcal{M}}$, where ∇f is the gradient of f. Let $d_{\mathcal{M}}(\mathbf{x}, \mathbf{y})$ be l, and let the geodesic curve that links between $\mathbf{x}$ and $\mathbf{y}$ on $\mathcal{M}$ be denoted as $c(t)$, which is parameterized by the arc length of the curve and satisfies $c(0) = \mathbf{x}$ and $c(l) = \mathbf{y}$.

Then $f(\mathbf{y})$ can be written as

$$f(\mathbf{y}) = f(\mathbf{x}) + \int_0^l df(c'(t))dt = f(\mathbf{x}) + \int_0^l \langle \nabla f(c(t)), c'(t) \rangle_{\mathcal{M}} dt \tag{3.30}$$

Then by Schwartz inequality, and use the fact that $||c'(t)|| = 1$ (because $c(t)$ is parameterized by its arc length)

$$\langle \nabla f(c(t)), c'(t) \rangle \leq ||\nabla f(c(t))|| \cdot ||c'(t)|| = ||\nabla f(c(t))||$$

Also, by performing Taylor expansion at $t = 0$, where $c(0) = \mathbf{x}$, we can estimate ∇f along the curve $c(t)$:

$$||\nabla f(c(t))|| = ||\nabla f(\mathbf{x})|| + O(t)$$

Thus, distance between $f(\mathbf{y})$ and $f(\mathbf{x})$ can be estimated as:

$$|f(\mathbf{y}) - f(\mathbf{x})| \leq l \cdot ||\nabla f(\mathbf{x})|| + O(l^2)$$

We can take $0 < l \ll 1$ for $\mathbf{x}$, $\mathbf{y}$ that are close together Then the above can be expressed as

$$|f(\mathbf{y}) - f(\mathbf{x})| \leq l \cdot ||\nabla f(\mathbf{x})|| + o(l) \tag{3.31}$$

where O and o are used in the infinitesimal sense, and $l = d_{\mathcal{M}}(\mathbf{x}, \mathbf{y})$. Under the current settings for $\mathcal{M}$, the following is shown as in Bousquet et al. (2004) Belkin and Niyogi (2005):

$$d_{\mathcal{M}}(\mathbf{x}, \mathbf{y}) = ||\mathbf{x} - \mathbf{y}||_{\mathbb{R}^D} + o(||\mathbf{x} - \mathbf{y}||_{\mathbb{R}^D}) \tag{3.32}$$

Thus

$$|f(\mathbf{x}) - f(\mathbf{y})| \leq ||\nabla f(\mathbf{x})|| \cdot d_{\mathcal{M}}(\mathbf{x}, \mathbf{y}) + o(d_{\mathcal{M}}(\mathbf{x}, \mathbf{y}))$$

as claimed in the Proposition. □

Now it is clear that if $||\nabla f(\mathbf{x})||$ is small, then $f(\mathbf{x})$ is close to $f(\mathbf{y})$ for neighboring $\mathbf{x}$ and $\mathbf{y}$. By requiring that the average of $||\nabla f(\mathbf{x})||$ at all $\mathbf{x}$ to be small, we can form the following optimization problem for finding the map f:

$$\min_{||f||_{L_2(\mathcal{M})}=1} \int_{\mathcal{M}} ||\nabla f(\mathbf{x})||^2 \tag{3.33}$$

To write this minimization problem in terms of the Laplace-Beltrami operator, we notice that the above integration can be expressed using inner product: $\int_{\mathcal{M}} ||\nabla f(\mathbf{x})||^2 = \int_{\mathcal{M}} \langle \nabla f(\mathbf{x}), \nabla f(\mathbf{x}) \rangle$. By Stoke's theorem, $-div$ and ∇ are formally adjoint operators. Then,

$$\int_{\mathcal{M}} \langle \nabla f(\mathbf{x}), \nabla f(\mathbf{x}) \rangle = -\int_{\mathcal{M}} f(\mathbf{x}) div \nabla f(\mathbf{x})$$

Use the fact that the Laplace-Beltrami operator is defined as

$$\mathcal{L} f = -div \nabla f$$

we have the following expression:

$$\int_{\mathcal{M}} ||\nabla f(\mathbf{x})||^2 = \int_{\mathcal{M}} f \mathcal{L} f \tag{3.34}$$

Thus the minimization problem (3.33) becomes:

$$\min_{||f||_{L_2(\mathcal{M})}=1} \int_{\mathcal{M}} f \mathcal{L} f \tag{3.35}$$

This can be seen as a continuous analogy of (3.27) and (3.28).

From (3.34), we conclude the Laplace-Beltrami operator is positive semidefinite and thus all eigenvalues of $\mathcal{L}$ have to be nonnegative. It is proved that the spectrum of $\mathcal{L}$ on a compact manifold is discreteRosenberg (1997). We can denote these eigenvalues as $0 = \lambda_0 \leq \lambda_1 \leq \cdots$, and the corresponding eigenfunctions as $f_0, f_1, \cdots$. The function that maps all points to a same representation is an eigenfunction of $\mathcal{L}$ that corresponds to the eigenvalue $\lambda_0 = 0$. We call it f_0, and it should be avoided as it does not differentiate different data points. Then, analogic to the case in (3.27), the minimizer f of (3.35) has to be found among those orthogonal to the constant function and it follows that the eigenfunction f_1 is the minimizer of (3.35). In addition, if we are looking for a p-dimensional map $f : \mathcal{M} \to \mathbb{R}^p$ under the same problem settings, it can be found as

$$\forall \mathbf{x} \in \mathcal{M}, \quad f(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \cdots, f_p(\mathbf{x})) \tag{3.36}$$

where f_i is the eigenfunction that corresponds to eigenvalue λ_i.

3.2.4.3 Weight Matrix Suggested by the Heat Kernel

Laplace-Beltrami operator is related to the heat flow on a manifold $\mathcal{M}$ through the heat equation that is defined as

$$(\frac{\partial}{\partial t} + \mathcal{L})u = 0$$
$$u(\mathbf{x}, 0) = f(\mathbf{x})$$

where $u(\mathbf{x}, t)$ is the heat distribution at point $\mathbf{x}$ and time t, and $f(\mathbf{x})$ is the heat distribution at time $t = 0$.

Let $H_t(\mathbf{x}, \mathbf{y})$ be the Green's function for the above partial differential equation, then the solution can be written as $u(\mathbf{x}, t) = \int_{\mathcal{M}} H_t(\mathbf{x}, \mathbf{y}) f(\mathbf{y})$. Using this solution expression, at time $t = 0$ we derive:

$$\mathcal{L}f = \mathcal{L}u(x, 0) = -\left(\frac{\partial}{\partial t}\int_{\mathcal{M}} H_t(\mathbf{x}, \mathbf{y}) f(\mathbf{y})\right)_{t=0} \tag{3.37}$$

Although the heat kernel on a given m-dimensional manifold $\mathcal{M}$ is not easy to find, it is shown in Rosenberg (1997) that, in an appropriate coordinate system, when $\mathbf{y}$ is close to $\mathbf{x}$ and t is small, the heat kernel can be approximated as

$$H_t(\mathbf{x}, \mathbf{y}) \approx (4\pi t)^{-\frac{m}{2}} \exp\left(-\frac{||\mathbf{x} - \mathbf{y}||^2}{4t}\right)$$

Notice that H_t tends to a Dirac's δ-function as t tends to 0 Belkin and Niyogi (2003), that is,

$$\lim_{t \to 0} \int_{\mathcal{M}} H_t(\mathbf{x}, \mathbf{y}) f(\mathbf{y}) = f(\mathbf{x})$$

Then we can approximate $\mathcal{L}f$ from (3.37) when t small:

$$\mathcal{L}f(\mathbf{x}) \approx -\frac{1}{t}\left[(4\pi t)^{-\frac{m}{2}} \int_{\mathcal{M}} \exp\left(-\frac{||\mathbf{x} - \mathbf{y}||^2}{4t}\right) f(\mathbf{y}) - \lim_{t \to 0} \int_{\mathcal{M}} H_t(\mathbf{x}, \mathbf{y}) f(\mathbf{y})\right]$$

That is,

$$\mathcal{L}f(\mathbf{x}) \approx \frac{1}{t}\left[f(\mathbf{x}) - (4\pi t)^{-\frac{m}{2}} \int_{\mathcal{M}} \exp\left(-\frac{||\mathbf{x} - \mathbf{y}||^2}{4t}\right) f(\mathbf{y})\right] \tag{3.38}$$

We can approximate the above expression using a selection of data points $\{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\}$ distributed on $\mathcal{M}$:

$$\mathcal{L}f(\mathbf{x}_i) \approx \frac{1}{t}\left[f(\mathbf{x}_i) - (4\pi t)^{-\frac{m}{2}} \sum_{\mathbf{x}_j \in \{\mathbf{y}:\, 0 < \|\mathbf{x}_i - \mathbf{y}\| < \epsilon\}} \exp\left(-\frac{||\mathbf{x}_i - \mathbf{x}_j||^2}{4t}\right) f(\mathbf{x}_j)\right] \tag{3.39}$$

This can be considered as suggestions for defining the weight matrix that used to define graph Laplacian L. (Here t is a global and does not affect the eigenvectors of L.) As mentioned in Belkin and Niyogi (2003), the coefficient $\alpha := (4\pi t)^{-\frac{m}{2}}$ is hard to determine since the dimension m of the given manifold is usually unknown. However, it can be determined by noticing that the application of $\mathcal{L}$ on any constant function is 0. That is, α can be determined from (3.39) with f being a constant function:

$$\alpha = \left(\sum_{\mathbf{x}_j \in \{\mathbf{y}:\, 0<\|\mathbf{x}_i-\mathbf{y}\|<\epsilon\}} \exp\left(-\frac{||\mathbf{x}_i - \mathbf{x}_j||^2}{4t}\right) \right)^{-1} \tag{3.40}$$

Expressions in (3.39) and (3.40) suggest a method for defining the weight matrix $W = (w_{ij})$:

$$w_{ij} = \begin{cases} \exp\left(-\frac{\|\mathbf{x}_i-\mathbf{x}_j\|^2}{4t}\right) & \text{if } \|\mathbf{x}_i - \mathbf{x}_j\| < \epsilon \\ 0 & \text{otherwise} \end{cases}$$

3.2.4.4 Discrete Operators that Approximate the Laplace-Beltrami Operator

The relationship between Laplace-Beltrami operator and the graph Laplacian is further studied and it has been shown that under certain conditions the graph Laplacian on discrete data points converges to the Laplace-Beltrami operator on the manifold, on which the data points reside Belkin and Niyogi (2005) Singer (2006). Notice that although such relationship provides support for using spectral clustering methods, the required conditions for the convergence results discussed in Belkin and Niyogi (2005) Singer (2006) usually are not satisfied by an arbitrary given dataset.

On the other hand, the graph Laplacian is not the only discrete operator that can be related to the Laplace-Beltrami operator. Research has been done to find other types of discrete operators that can preserve as many properties of the Laplace-Beltrami operator as possible Bronstein et al. (2008). It is proved in Wardetzky et al. (2007) that theoretically, there is no ideal discrete operators that can preserve all properties listed as in Bronstein et al. (2008). The practical aspect of this is, there exists a large variety of discrete operators that can preserve several out of all these properties satisfying the needs of specific application problems Wardetzky et al. (2007).

3.3 Applications

Spectral clustering methods are used in various aspects of applications including image segmentation, shape recognition, protein classification and so on. Direct applications of spectral clustering methods are made in solving such problems in the unsupervised setting. Furthermore, the spectral clustering itself can be learned through optimizations base on the given training sets to generate more accurate results according to the nature of specific problems. This section is devoted to the discussion of several applications of spectral clustering methods.

3.3.1 Image Segmentation and Shape Recognition

Image segmentation can be considered as a perceptual grouping problem in vision. Compared to the human ability of recognizing and separating different upfront objects from the background in images, computers rely on clustering algorithms to detect different regions in digital images. Image segmentation is achieved by treating pixels in a given image as the dataset and then clustering the pixels with respect to image properties at each pixel such that pixels in a same group have similar image properties and pixels in different groups have dissimilar image properties. Examples of such clustering include clustering the pixels by their intensities or colors, by their positions, and by how likely there exist contour edges between pairs of pixels . Shape recognition can be considered as a classification problem which determines whether the detected structures in a image match a given shape. The corresponding datasets in shape recognition problems depend on the desired shapes and each data point can represent a combination of image properties. Spectral clustering methods help to solve the problem of shape recognition by first constructing the similarity matrix through a function that measures the possibilities of any pair of data points being a part of the desired shape, and then clustering the dataset using this similarity matrix. This section discusses some applications of spectral clustering in image segmentation and shape recognition.

3.3.1.1 Direct Applications in Unsupervised Learning

The normalized spectral clustering based on the first few eigenvectors of L_{sym} as defined in (3.13) can be applied to solve image segmentation problems in the unsupervised sense Shi and Malik (2000). A single digital image can be segmented based on the brightness, colors, image textures, etc. Motion segmentations Shi and Malik (2000) Shi and Malik (1998) can be done on a sequence of images. The image segmentations are realized by incorporating related information in the weight matrix W. Some methods of defining the weight matrix are proposed in Shi and Malik (2000), and they are described below.

In a given digital image that has $N \times N$ pixels, each pixel can be considered as a data point whose properties is represented by a d-dimensional vector. Then the dataset derived from the image consists of N^2 d-dimensional data points. A graph G can be constructed using the data points as vertices. The choices of the weight on the edge linking the ith and the jth data points made in Shi and Malik (2000) are the following:

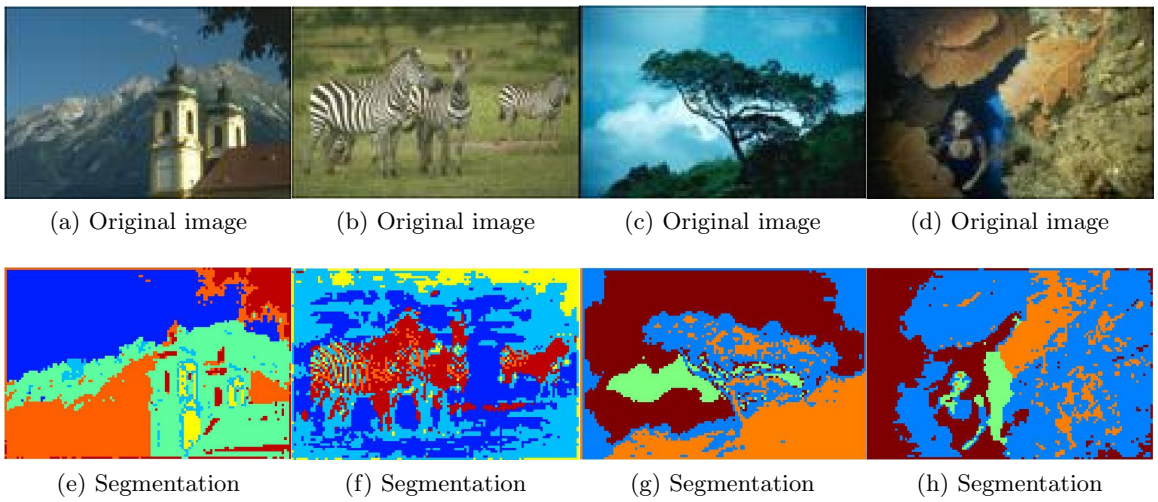
$$w_{ij} = \exp\left(-\frac{\|\mathbf{F}(i)-\mathbf{F}(j)\|_2^2}{\sigma_I}\right) \times \begin{cases} \exp\left(-\frac{\|\mathbf{X}(i)-\mathbf{X}(j)\|_2^2}{\sigma_X}\right) & \text{if} \|\mathbf{X}(i)-\mathbf{X}(j)\| < r \\ 0 & \text{otherwise} \end{cases} \tag{3.41}$$

where the spatial location of the ith data point is $\mathbf{X}_i$, and its other types of properties such as colors, intensities, or texture information are represented as $\mathbf{F}_i$, which can be defined as follows:

1. $\mathbf{F}_i = I(i)$, where $I(i)$ is the intensity value at the i-th data point. In this case the spectral clustering method segments the image by brightness and locations of pixels.

2. $\mathbf{F}_i = [v, v \cdot s \cdot \sin(h), v \cdot s \cdot \cos(h)](i)$, where h, s, v are the HSV values. In this case the spectral clustering method segments the image by colors and locations of pixels.

3. $\mathbf{F}_i = [|I * f_1|, \cdots, |I * f_n|](i)$, where f_i is the DOOG filters at various scales and orientations as in Malik and Perona (1990). In this case, the spectral clustering method segments the image by the texture information and locations of pixels.



(a) Original image (b) Original image (c) Original image (d) Original image

(e) Segmentation (f) Segmentation (g) Segmentation (h) Segmentation

Figure 3.1: From (a) to (d): original images. From (e) to (h): segmentation results obtained by using normalized spectral clustering algorithm.

Then by applying the normalized spectral clustering algorithm introduced in section 3.2.1.1, we can derive the desired image segmentation.

3.3.1.2 Learning Spectral Clustering for Recognition and Segmentation

Spectral clustering algorithms that target at solving specific problems need to be designed with suitable weight matrix. Learning processes that devote to train the weight matrices such that the spectral clustering methods are more likely to generate good results have been developed Cour and Shi (2004) Bach and Jordan (2004). The main procedure of this learning process is to optimize the weight matrix such that when one applies spectral clustering to any image in the training set, the first few eigenvectors are not very different from the indicator vectors of hand selected clusters in the image. This procedure is formulated in Cour and Shi (2004) in terms of minimizing an objective error function, and a learning algorithm was proposed there to find the appropriate weight matrix using gradient descent technique. The authors were able to calculate the derivatives of the *Ncut* eigenvectors in exact analytic form, and thus were able to develop the theoretical justification of their training procedure. The resulting weight matrix is able to memorize and retrieve the desired shapes from noisy background in the shape recognition problem. The justification of this technique and the details of the procedure are

given in Cour and Shi (2004) and summarized as follows.

For a given training image I with n pixels, let $X^*(I) \in \mathbb{R}^n$ be the ground truth label vector for segmentation or recognition. The components in $X^*(I)$ are in the set of $\{+1, -1\}$, which indicate two segments of the image in image segmentation problem, or indicate whether the related pixels are parts of the desired shape in the shape recognition problem. The error energy function is defined below to measure whether a weight matrix has been defined appropriately to give rise to the ground truth label vector:

Definition 11. *Let W be the weight matrix currently under consideration. Let $X_p[W], \lambda_p$ be the p^{th} largest eigenvector and eigenvalue of $WX = \lambda DX$, where L and D are defined as in (3.8) and (3.7), and $X_p[W]$ has unit length. Define $X_{ncut}[W] = X_2[W]$ for $W \in S_n^{2,X^*(I)}$, which is a certain subset of S_n, the set of symmetric matrices in $\mathbb{R}^{n\times n}$. This subset avoids singularities when defining $X_{ncut}[W]$ uniquely. The one-target energy function is defined as:*

$$\mathcal{E}(W, I) = \frac{1}{2}\|X_{ncut}[W(I)] - X^*(I)\|^2, \quad \text{for } W \in S_n^{2,X^*(I)} \tag{3.42}$$

And the multi-target energy function is defined as:

$$\mathcal{E}(W, I) = \sum_I \mathcal{E}(W, I), \quad \text{for } W \in \bigcap_I S_n^{2,X^*(I)} \tag{3.43}$$

Here, $S_n^{p,Y} = \{W \in S_n \ : \ D > 0, \ \lambda_p \text{ single}, \ ker(W - \lambda_p D) \bigcap Y^{\perp} = \{0\}\}$. This definition also ensures that $Y^T X_p \neq 0$. X_p are defined uniquely by requiring $Y^T X_p > 0$.

The one-target error energy function has the following property which justifies the minimization procedure:

Proposition 12 ($\mathcal{E}(W, I)$ **has no local minimum**). *The one-target error energy function has all its local minimum in $S^{2,X^*} \bigcap \{W : \lambda_2(W) \neq -1\}$ equal to the global minimum, 0.*

The proof can be found in the appendix of Cour and Shi (2004).

The minimization of the error energy function over the weight matrix can be done by using the gradient descent technique:

$$W := W - \eta \frac{\partial \mathcal{E}}{\partial W} \tag{3.44}$$

Expressed in a continuous time partial differential equation, the above technique can be considered as:

$$\dot{W} = -\frac{\partial \mathcal{E}}{\partial W} = -\frac{\partial \mathcal{E}}{\partial X_{ncut}[W]} \frac{\partial X_{ncut}[W]}{\partial W} \tag{3.45}$$

Then the problem turns out to be showing that an exact analytic form of $\frac{\partial X_{ncut}[W]}{\partial W}$ exists and the PDE converges.

First, the following theorem gives the analytic expression of the derivative of *Ncut* eigenvectors with respect to the parameter t along a C^1 path of the weight matrix $W(t)$.

Theorem 13 (**Derivative of Ncut eigenvectors**). *The map* $W \to (X_p, \lambda_p)$ *is* C^∞ *over* $S_n^{p,Y}$, *and we can express the derivatives over any* C^1 *path* $W(t)$ *as:*

$$\frac{dX_p[W(t)]}{dt} = -(W - \lambda_p D)^\dagger (W' - \lambda_p D' - \frac{d\lambda_p}{dt} D) X_p \tag{3.46}$$

$$\frac{d\lambda_p}{dt} = \frac{X_p^T (W' - \lambda_p D') X_p}{X_p^T D X_p} \tag{3.47}$$

where $A^\dagger$ is the pseudo-inverse of A.

The proof of above theorem can be found in the appendix of Cour and Shi (2004), where the implicit theorem can be used to show $X_p[W]$ is C^∞ and differentiating the equation $WX_p = \lambda_p D X_p$ gives expressions in (3.46). Also, left multiplying by X_p^T gives (3.47).

Then the authors in Cour and Shi (2004) introduced a term $Y = -(W - \lambda_2 D)^\dagger (X_{ncut} - X^*(I))$ to reduce the computation complexity from $O(n^3)$ to $O(n^2)$, and they induced the following expressions:

$$\frac{\partial \mathcal{E}}{\partial W_{ij}} = X_{ncut}^i Y_j + X_{ncut}^j Y_i - \lambda_2 (X_{ncut}^i Y_i + X_{ncut}^j Y_j) - \lambda'_{2ij} Y^T D X_{ncut} \tag{3.48}$$

and

$$\lambda'_{2ij} = (2X^i_{ncut}X^j_{ncut} - \lambda_2({X^i_{ncut}}^2 + {X^j_{ncut}}^2))/X^T_{ncut}DX_{ncut} \tag{3.49}$$

The above expressions provided the gradient descent technique with updating formulas. In the appendix of Cour and Shi (2004), the authors also showed that following the gradient descent path, the one-target energy function converges to 0 exponentially fast, and this behavior has also been observed empirically for multi-target energy function. Further, the convergence of the weight matrix together with the convergence of the energy function during the minimization procedure by gradient descent was shown by the following proposition in Cour and Shi (2004):

Proposition 14 (**Exponential convergence of the one-target learning rule to a global minimum**). *The PDE* $\dot{W} = -\frac{\mathcal{E}}{\partial W}$ *either converges to a global energy minimum* W_∞*, or escapes any compact set* $K \subset S_n^{2,X^*}$*. In the first case,* $\mathcal{E}(W) \to 0$ *exponentially.*

Thus, a learning procedure to achieve suitable weight matrices based on a given training set can be formulated as follows:

Spectral Learning Algorithm for Segmentation and Recognition Cour and Shi (2004)

1. Initialize a random weight matrix W_0 with preferable properties: W_0 is with small variance and large eigengap.

2. Repeat step 3 and 4 until $\sum_I \mathcal{E}(W, I) <$ threshold

3. Compute $\bar{X_2}$, λ_2 as the second smallest eigenvector and eigenvalue of $D^{-1/2}W(I)D^{-1/2}$. Then define:
$$X_{ncut} = D^{-1/2}\bar{X_2}/\|D^{-1/2}\bar{X_2}\| \cdot sign(X^*(I)^T D^{-1/2}\bar{X_2})$$

4. Update the weight matrix by gradient descent: $W(I) := W(I) - \eta\frac{\partial\mathcal{E}(W,I)}{\partial W(I)}$ with (3.48), and compute $\mathcal{E}(W, I) = \frac{1}{2}\|X_{ncut} - X^*(I)\|^2$.

The above algorithms were shown to be efficient and effective in solving data clustering problem, geometric shape detection problem and multiple shape recognition problems through

examples provided in Cour and Shi (2004). The geometric shape recognition example showed there aimed at recognizing rectangular shapes in images by learning from a training set. For a given image, the corresponding dataset is the set of all detected edges $V = (Edge_{x_i,\theta_j})$ with x_i representing the edge location and θ_j representing the edge direction. The weight between two edges can be written as a function $W(Edge_{x_i,\theta_j}, Edge_{x_{i'},\theta_{j'}}) = f(x_{i'} - x_i, \theta_{j'} - \theta_j)$, which can be learnt by the training set. The result obtained by applying the above algorithm successfully recognized rectangular shapes in the testing set and augmented the desired shapes by reducing the noise in the background. The multiple shape recognition example shown in Cour and Shi (2004) targeted at memorizing and retrieving multiple desired shapes in solving shape recognition problems. The above algorithm was shown successful in recognizing 10 different targets in a related example while removing irrelevant pixels. More details and related applications were provided in Cour et al. (2005b) and the above algorithms were applied to real world images to recognize rectangular shapes.

3.3.1.3 Segmentation by Multiscale Graph Decomposition

Image segmentation problems are difficult to solve partly because the high computation complexity when one uses each pixel as a data point. Noisy background and faint contour edges of upfront objects in an image also bring difficulties. Algorithms have been designed to overcome such difficulties by using various multiscale techniques. The authors of Cour et al. (2005a) argued that in order to design a reasonable and effective multiscale method for segmentation, three issues have to be addressed: (1). how to enhance the faint contour edges; (2). how to conclude from regional information across multiple scales to provide information on textures or shapes of larger regions; (3). how to propagate the local grouping information of multiscale regions to achieve coherent segmentations. The authors justified and designed an algorithm that targets the last question through a parallel segmentation of regions across multiple scales.

They first designed the weights between two pixels by combining the information of their spatial locations, intensities and intervening contours, then investigated the statistics of graph

weights across a set of images. The conclusions implied that long range graph connections have more redundant information and can be compressed. Then for a given image, the multiscale weight matrix can be expressed as a combination of weight matrices over different scales:

$$W = W_1 + W_2 + \cdots + W_S \tag{3.50}$$

where each W_s being the affinity matrix between pairs of pixels with a fixed range of spatial separation $W_s(i,j) \neq 0$ only if $G_{r,s-1} < r_{ij} \leq G_{r,s}$, $G_{r,s}$, $s \in \{1, 2, \cdots, S\}$ are scales of distances. The first matrix W_1 is constructed by taking each pixel as a graph node, but two nodes are connected only if they are within distance r apart by a graph edge. The next matrix W_2 takes into account the information at a larger scale, it is constructed as the affinity matrix of sampled nodes that are connected by graph edges of length at least $r+1$ and at most $2r+1$. Recursively one can define weight matrices at larger scales. At scale s, W_s is constructed from the sampled nodes that are at least $(2r+1)^{s-1}+1$ distance apart and at most within distance $(2r+1)^s$ on the original image grid. Compared to the weight matrix that can be constructed by taking into account every pair of pixels, the above compression is not perfect but can preserve significant information and reduce computational complexity. Demonstration of reconstruction from the compressed series of multiscale matrices was shown in Cour et al. (2005a).

A parallel procedure was proposed in Cour et al. (2005a) to solve the segmentation problem simultaneously across different scales. Let $X_s \in \{0,1\}^{N_s \times K}$ be the indicator matrix of the partition at scale s, where $X_s(i,k) = 1$ if and only if the node i in the set of nodes I_s at the scale s belongs to the k-th cluster. The multiscale partition matrix $\mathbf{X}$ and the multiscale affinity matrix $\mathbf{W}$ are defined as follows:

$$\mathbf{X} = \begin{pmatrix} X_1 \\ \vdots \\ X_S \end{pmatrix}, \quad W = \begin{pmatrix} W_1^c & & 0 \\ & \ddots & \\ 0 & & W_S^c \end{pmatrix} \tag{3.51}$$

where the c in the super index indicates these matrices are obtained by compression and their nodes are sampled from the original image.

In order to propagate from small scale to large scale and keep the results coherent and consistent, additional requirements are posed to the spectral clustering instead of using the multiscale affinity matrix alone. That is, to keep consistence, the proposed algorithm requires that for any node i in layer I_{s+1}, $X_{s+1}(i) = \frac{1}{|\mathcal{N}_i|}\sum_{j\in\mathcal{N}_i} X_s(j)$. It will be convenient for us to define the following **cross scale interpolation matrix** between nodes in layer I_s and I_{s+1}:

$$C_{s,s+1}(i,j) = \begin{cases} \frac{1}{|\mathcal{N}_i|} & \text{if } j \in \mathcal{N}_i \\ 0 & \text{otherwise} \end{cases} \tag{3.52}$$

Then the **cross scale constraint matrix** can be defined as:

$$C = \begin{pmatrix} C_{1,2} & -I_2 & & 0 \\ & \ddots & \ddots & \\ 0 & & C_{S-1,S} & -I_S \end{pmatrix} \tag{3.53}$$

Thus, the **cross scale segmentation constraint equation** can be formulated as follows:

$$CX = 0 \tag{3.54}$$

To combine the above constraints with the spectral clustering methods, the multiscale segmentation algorithm presented in Cour et al. (2005a) can be summarized as follows:

Multiscale segmentation criterion:

$$\text{maximize } \epsilon(X) = \frac{1}{K}\sum_{m=1}^{K}\frac{X_m^T W X_m}{X_m^T D X_m} \tag{3.55}$$

subject to constraint

$$CX = 0, \;\; X \in \{0,1\}^{N^*\times K}, \;\; X1_K = 1_{N^*} \tag{3.56}$$

where $N^* = \sum_s N_s$.

The above can be transformed after algebraic computations to the following problem:

$$\text{maximize } \epsilon(Z) = \frac{1}{K} tr(Z^T W Z) \tag{3.57}$$

subject to constraints

$$CZ = 0, \quad Z^T DZ = I_K \tag{3.58}$$

Efficient computational techniques and a summarized algorithm for solving the above constraint optimization problem were presented in Cour et al. (2005a). Demonstration of its ability of reducing computation complexity and some successful applications of this algorithm were also included there. Fig. 3.2 gives some examples of image segmentation by applying the multi-scale spectral method introduced in Cour et al. (2005a). These examples demonstrate that this spectral segmentation method is able to segment the given images based on similar colors and also preserve the delicate details.

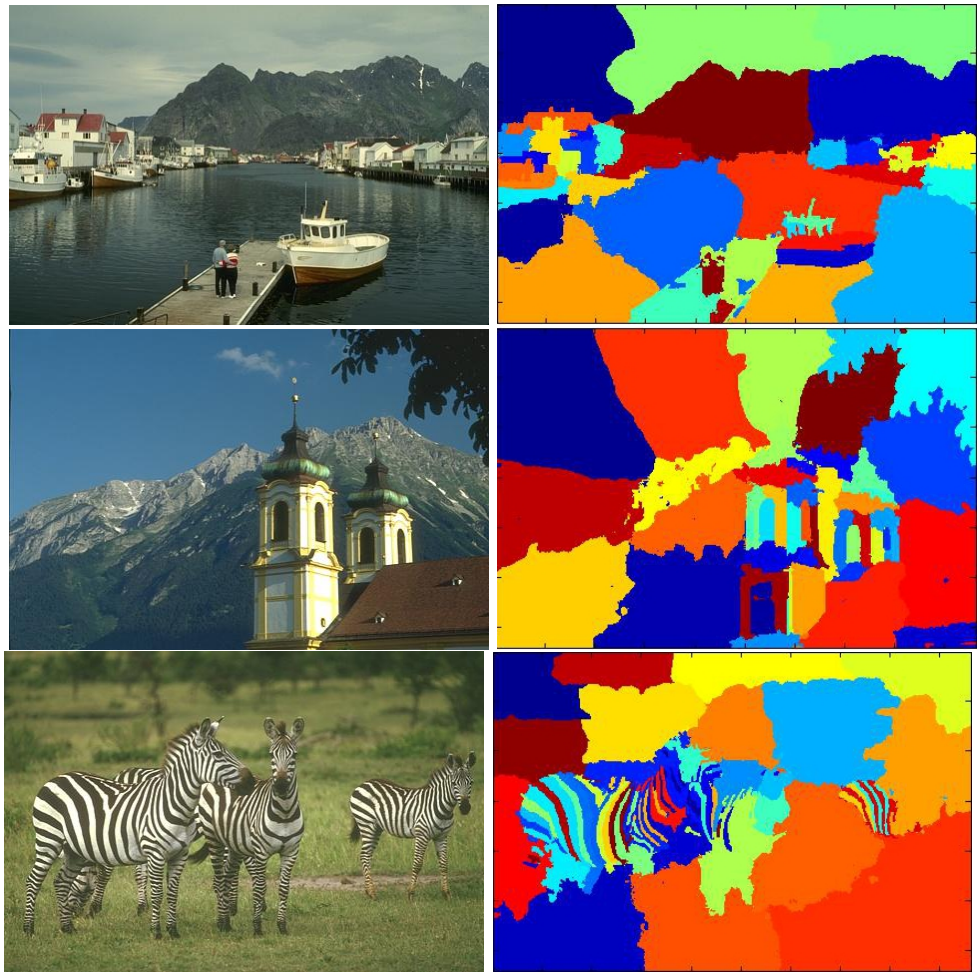


Figure 3.2: Image segmentation results given by multi-scale spectral segmentation method Cour et al. (2005a). left column: original images. right column: segmentation results obtained by requiring the number of clusters to be 40, 45, and 40 respectively.

3.4 Learning Spectral Clustering Methods

The construction of a suitable weight matrix for a specific clustering problem is crucial for spectral clustering methods to generate good results. Methods designed to learn the affinity functions used in the weight matrix have been proposed and proved to be efficient. Examples include learning spectral clustering in the image segmentation setting as described in the previous section. In the following, we discuss another perspective of learning spectral clustering.

3.4.1 Learning Spectral Clustering

In the setting of spectral clustering, for a given dataset $X \subset \mathbb{R}^d$ consists of P points, one can construct a graph G using the P data points as the set of nodes V and design a weight matrix W. A scheme is proposed in Bach and Jordan (2004)to learn the suitable weight matrix by first designing a cost function that measures how unlikely a weight matrix may induce a partition. Then the minimization of the cost function can be understood in two ways. On one hand, the minimization of the cost function with respect to the choices of the partition for a fixed weight matrix generates a clustering method. On the other hand, the minimization with respect to the weight matrix provided with the known clustering results give rise to the appropriate design of the weight matrix. The arguments made in Bach and Jordan (2004) are described as follows.

If the graph $G = (V, E)$ constructed based on the dataset X can be partitioned into R disjoint clusters $A = \{A_r\}_{r=1}^R$ with $\bigcup_r A_r = V$, then the R-way normalized cut is defined as

$$C(A, W) = \sum_{r=1}^{R} \Big(\sum_{i \in A_r, j \in V \setminus A_r} W_{ij} \Big) / \Big(\sum_{i \in A_r, j \in V} W_{ij} \Big) \tag{3.59}$$

which is the total sum of the ratios of any cluster's inter-cluster affinity to its total affinity. The minimization of $C(A, W)$ with respect A gives a partition of X. We consider $e_r \in \{0, 1\}^P$ as the indicator of the r-th cluster and an equivalent representation of the partition A. Then the normalized cut can be written by

$$C(e, W) = \sum_{r=1}^{R} e_r^T (D - W) e_r / (e_r^T D e_r) \tag{3.60}$$

The following proposition shows the relationship between the normalized cut and the eigenvalue problem:

Proposition 15 (Bach and Jordan (2004)). *The R-way normalized cut is equal to* $R - trY^T D^{-1/2} W D^{-1/2} Y$ *for any matrix* $Y \in \mathbb{R}^{P \times R}$ *such that*
(a). the columns of $D^{-1/2}Y$ *are piecewise constant with respect to the clusters and*
(b). Y *has orthonormal columns (i.e.* $Y^T Y = I$*)*

It has been shown that by dropping the constraint (a) in the minimization of $C(e, W)$, the relaxed optimization problem leads to the classical lower bound on the optimal normalized cut Chan et al. (1993) Zha et al. (2001). The following result relates this optimization to an eigenvalue problem, which can be more easily solved:

Proposition 16 (Bach and Jordan (2004)). *The maximum of* $trY^T D^{-1/2} W D^{-1/2} Y$ *over matrices* $Y \in \mathbb{R}^{P \times R}$ *such that* $Y^T Y = I$ *is the sum of the R largest eigenvalues of* $D^{-1/2} W D^{-1/2}$*. It is attained at all* Y *of the form* $Y = UB_1$ *where* $U \in \mathbf{P} \times \mathbf{R}$ *is any orthonormal basis of the R-th principal subspace of* $D^{-1/2} W D^{-1/2}$ *and* B_1 *is an arbitrary rotation matrix in* $\mathbb{R}^{P \times R}$*.*

Although the above relaxed optimization problem is relatively easy to solve, the solution Y may not have the property that every column of Y is piecewise constant with respect to the partition, i.e. there may not exist a matrix Λ such that $Y = E\Lambda$ where $E = (e_1, e_2, \cdots, e_R)$. Thus, to recover the requirement that every column of Y is piecewise constant with respect to the partition, one can approximate the solution Y using matrix which satisfy this requirement. Because the solution Y and such piecewise constant matrix are both unique up to some rotations, then the comparison between them only makes sense as comparison between the subspaces spanned by them. One way is to compare the orthogonal projection operators on these spaces by computing the Frobenius norm between UU^T and $\Pi_0 = \Pi_0(W, e) = \sum_r D^{1/2} e_r e_r^T D^{1/2} / (e_r^T D e_r)$. Since the goal is to obtain minimize the difference between Y and the piecewise constant matrix,

thus is to minimize the following designed cost function:

$$J(W,e) = \frac{1}{2}||UU^T - \Pi_0||_F^2 \tag{3.61}$$

which can be calculated as

$$J(W,e) = R - trUU^T\Pi_0 = R - \sum_r e_r^T D^{1/2}UU^T D^{1/2}e_r/(e_r^T De_r) \tag{3.62}$$

From the above analysis, it is clear that the minimization of the cost function $J(W,e)$ with respect to partition e yields a partition that the most likely has the minimum cut on the graph weight by W. And the authors of Bach and Jordan (2004) proved that it can be considered as a weight distortion measure.

The following theorem further validates a weighted K-means procedure for the minimization of the cost function with respect to the partition.

Theorem 17. *Let W be an affinity matrix and let $U = (u_1, u_2, \cdots, u_P) \subset \mathcal{R}^{R\times P}$ be an orthonormal basis of the R-th principal subspace of $D^{-1/2}WD^{-1/2}$. Then for any partition e (which can also be represented as A in terms of the partition of nodes),*

$$J(W,e) = \min_{(\mu_1,\cdots,\mu_R)\in\mathcal{R}^{R\times R}} \sum_r \sum_{p\in A_r} d_p||u_p d_p^{-1/2} - \mu_r||^2 \tag{3.63}$$

This algorithm can be summarized as follows:

On the other hand, if one chooses to minimize the cost function $J(W,e)$ with respect to the affinity matrix for one or several given partition(s), then the minimization will give rise to the best way to define the affinity matrix that is able to give the correct partitions.

First, given two partitions $e = (e_r)$ and $f = (f_s)$ with R and S clusters respectively, the difference between them can be computed as the following Bach and Jordan (2004):

$$d(e,f) = \frac{1}{2}||\sum_r \frac{e_r e_r^T}{e_r^T e_r} - \sum_s \frac{f_s f_s^T}{f_s^T f_s}||_F^2 = \frac{R+S}{2} - \sum_{r,s} \frac{(e_r^T f_s)^2}{(e_r^T e_r)(f_s^T f_s)} \tag{3.64}$$

Table 3.1: Learning Spectral Clustering Algorithm

Input:	a given affinity matrix $W \in \mathcal{R}^{P\times P}$.
Step 1:	Compute the first R eigenvectors U of $D^{-1/2}WD^{-1/2}$ where D is the degree matrix. Denote $U = (u_1, u_2, \cdots, u_P)$ and $d_p = D_{pp}$.
Step 2:	
	1. For all r, let $\mu_r = \sum_{p\in A_r} d_p^{1/2} u_p / \sum_{p\in A_r} d_p$
	2. For all p, assign p to A_r where $r = \arg\min_{r'} \|\|u_p d_p^{-1/2} - \mu_{r'}\|\|$
Step 3:	Repeat step 2 until the partition A is stationary.

It is pointed out in Bach and Jordan (2004) that the above measure is between zero and $\frac{R+S}{2} - 1$ and equals zeros if and only if $e = f$. If we consider the search for the best partition $e(W)$ for a fixed W as a generalized K-means method (as described before), the following theorem shows that if one performs this K-means method exactly, one can obtain an upper bound for the quality of the partition $e(W)$.

Theorem 18. *Let* $\eta = \max_p D_{pp} / \min_p D_{pp} \geq 1$. *If* $e(W) = \arg\min_e J(W, e)$, *then for all partitions* e, *we have* $d(e, e(W)) \leq 4\eta J(W, e)$.

To learn the weight matrix that is appropriate to use in spectral clustering when provided with one or more than one datasets where the exact clustering are known, the authors in Bach and Jordan (2004) used the weight matrix $W_{ij} = \exp\left[-(\mathbf{x}_i - \mathbf{x}_j)^T diag(\alpha)(\mathbf{x}_i - \mathbf{x}_j)\right]$ for simplicity and concreteness. Here $\alpha \in \mathbb{R}^F$ and it can scale the Euclidean distances in the weight matrix. This provides a parametric study of the learning procedure, while this procedure can be generalized to other types of weight matrices.

Assume that we are given N datasets D_n, $n = 1, 2, \cdots, N$, with $D_n \in \mathbb{R}^F$ and consists of data points x_{np}, $p = 1, 2, \cdots, P_n$. These training sets can be considered as the ones draw from a dataset D, which we hope to cluster. Let the known exact clustering of D_n be e_n, and let the weight matrix for each D_n be W_n. For each n, the target matrix can be denoted as $\Pi_0(e_n, \alpha)$.

Thus the total cost function can be written as:

$$H(\alpha) = \frac{1}{N} \sum_{n} F(W_n(\alpha), \Pi_0(e_n, \alpha)) + C||\alpha||_1 \tag{3.65}$$

where the l_1 penalty on the vector α helps to select sparse candidates of the weight matrix. The learning procedure is then formulated as a minimization process on $\alpha \in \mathbb{R}_+^F$, and the minimization by conjugate gradient with line search is suggested in Bach and Jordan (2004). The minimizer α can then be used in the weight matrix to cluster the dataset D. Results shown in Bach and Jordan (2004) demonstrated the superior performance of this learning process.

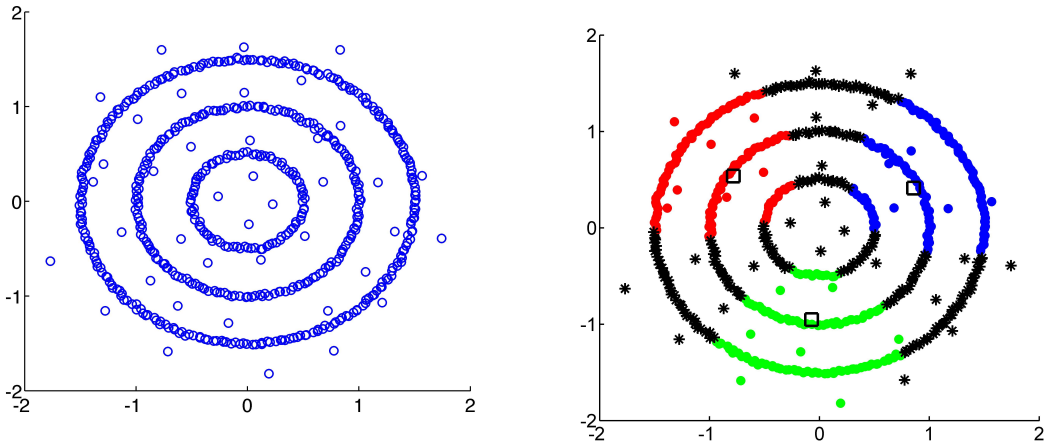
CHAPTER 4. Novel Data Clustering Method Fuzzy-RW

As described in previous chapters, fuzzy c-means method is popular because of its simplicity and efficiency, and the spectral clustering methods are also used broadly in applications because of their abilities to cluster on the lower dimensional space where the data points actually reside on. We designed a novel data clustering algorithm that combines the strength of spectral clustering methods with that of the fuzzy c-means method by making use of a family of special distances defined based on the random walks on the underlying data structure. In addition, a penalty term is designed in the algorithm to provide an efficient and accurate method to search for the centriods of clusters. For datasets that are perturbed by noise, we can also modify the algorithm by considering the local properties of datasets and redefining the random walks. Such strategy can be used in clustering under users' directional preferences or in clustering datasets with local neighborhood formed in some elongated shapes. For the latter case, we design a local principal component analysis operation to adaptively detect the local geometric properties and refine the random walk distances accordingly. The following sections are devoted to the description of this novel clustering algorithm. Specifically, the distances defined based on random walks is described in section 1, the optimization framework including the penalty term is given in section 2, the noise-reducing modification based on the local density is introduced in section 3, the clustering with users' directional preferences are given in section 4, finally the technique of automatic detections of the preferential directions for the random walks is described in section 5.

4.1 Distances Defined by Random Walks on the Graph

For a high dimensional dataset $X = \{\mathbf{x}_i\}_{i=1}^N \subset \mathbb{R}^p$, although the Euclidean distance or other types of distances defined in the space $\mathbb{R}^p$ are able to provide some insight on the relationships between data points in the high dimensional space, they fail to recognize the underlying geometry of the data structure when the dataset resides on a lower dimensional space which is embedded in $\mathbb{R}^p$. Thus for clustering datasets that reside on a lower dimensional manifold and embedded in a higher dimensional space, clustering methods that rely on the Euclidean distances between data points are often misleading. For example, in Fig.4.1, the dataset consists of three circles which we would like to interpret as three clusters, and the dataset is perturbed with noise data points. The application of FCM with the Euclidean distance will lead to an equal sized partition of the space where the data points scatter. In this example, a family of distances is preferred if these distances are capable of assigning small distances between data points located on a same circle. That is, favored distance measures are those intrinsic to the underlying data structures. Spectral clustering methods try to solve this problem by first projecting the high dimensional dataset onto a lower dimensional space which is spanned by the eigenvectors that capture the geometric properties of the dataset. Then the data points are clustered using some ordinary type of distance defined in the lower dimensional space (for example, Euclidean distance). Other types of dimension reduction methods, for example principal component analysis method, also attempt to explore the underlying data structure by projecting an original high dimensional dataset to a lower dimensional one while optimizing some desired properties of the dataset (in the case of PCA, this property is the total variance). Here, without projecting the high dimensional dataset, we explore the dataset structure by defining a family of distances that incorporates local data relationships into a global one through random walks on the graph constructed from the given dataset.

Graph that is constructed from a given dataset is convenient in representing similarities between data points. Let graph G be constructed from the dataset $X = \{\mathbf{x}_i\}_{i=1}^N \subset \mathbb{R}^p$, where $G = (V, E)$ consists of the set of vertices V and the set of edges E. V is taken as the given



(a) (b)

Figure 4.1: (a) Dataset consisting of three core clusters and a uniform distribution of outliers. This geometric configuration leads to clusters which are not linearly separable. (b) Output of the FCM algorithm applied to the data in (a). The squares correspond to cluster centroids.

dataset X (i.e. the ith node of V corresponds to $\mathbf{x}_i$) and E consists of all edges between any pair of data points. A weighted adjacency matrix $W_{N\times N}$ can be defined such that W_{ij} is the weight posed on the edge that links $\mathbf{x}_i$ and $\mathbf{x}_j$. In order to relate W to the probabilities of taking random walks on the graph G, we require that the definition of W_{ij} reflects the similarity between $\mathbf{x}_i$ and $\mathbf{x}_j$. One popular choice of W is the following

$$W_{ij} = \exp\left(-\frac{||\mathbf{x}_i - \mathbf{x}_j||^2}{\sigma}\right) \tag{4.1}$$

The advantages of the above choice of weights in the context of spectral clustering and dimensionality reduction are discussed extensively in Belkin and Niyogi (2003) although other choices of definitions are also used in literature Coifman and Lafon (2006) Higham et al. (2007). In the above definition (4.1), the distance $||\cdot||$ is the Euclidean distance, and the choice of "bandwidth" parameter σ belongs to an active research area Coifman and Lafon (2006).

The degree matrix D can be defined as

$$D = diag(D_{11}, D_{22}, \cdots, D_{NN}), \quad D_{ii} = \sum_{j=1}^{N} W_{ij} \tag{4.2}$$

where D_{ii}, being the total sum of similarities between $\mathbf{x}_i$ and other data points, can reflect how

"connected" the data point $\mathbf{x}_i$ is among the given dataset.

Then the following matrix P can be considered as the transition matrix for random walks on the graph G:

$$P = D^{-1}W \tag{4.3}$$

where

$$0 \leq P_{ij} \leq 1, \ \sum_{j=1}^{N} P_{ij} = 1 \tag{4.4}$$

and P_{ij} represents the probability of the random walk starting from the ith node reaches the jth node in one step. We can define the random walks on G using the above transition matrix P.

The random walk on the graph G can be denoted as $\{X_t\}_{t\geq 0}$. We can consider the following hitting time associated with the random walks:

$$\tau_j = \inf\{t \geq 0 \mid X_t = \mathbf{x}_j\} \tag{4.5}$$

which is the first time the random walk reaches the jth node.

The expected time of the random walk which starts at the ith node reaches the jth node can reflect the distance between them. By using the transition matrix, the expected time for $i \neq j$ can be computed as the following:

$$\begin{aligned} E_i[\tau_j] &= 1 \cdot P(X_1 = \mathbf{x}_j, X_0 = \mathbf{x}_i) + E_i[\tau_j, X_0 = \mathbf{x}_i, X_1 \neq \mathbf{x}_j] \\ &= P_{ij} + \sum_{k\neq j} (1 + E_k[\tau_j])\, P_{ik} \end{aligned}$$

And $E_i[\tau_i] = 0$.

Let $A = (E_i[\tau_j])_{i\neq j}$, $Q = (P_{il})_{i\neq j, l\neq j}$, and $R = (P_{ij})_{i\neq j}$, then $E_i[\tau_j]$ can be computed using the matrix and vector formation:

$$A = R + Q(\mathbf{1} + A) \tag{4.6}$$

where $\mathbf{1}$ is the $N-1$ column vector with every coordinate as 1. Thus, the expected time the random walk takes to reaching the jth node ($j \neq i$) starting from the ith node can be computed as the following:

$$(E_i[\tau_j])_{i\neq j} = A = (I-Q)^{-1}(R+Q\mathbf{1}) \tag{4.7}$$

And to derive a symmetric distance measure between every pair of data points, we consider the following as the commute distance between every $\mathbf{x}_i$ and $\mathbf{x}_j$:

$$T_{ij} = \frac{1}{2}(E_i[\tau_j] + E_j[\tau_i]) \tag{4.8}$$

Another type of distance induced by the random walk on the graph G can be described in terms of the probability of some specific process. Specifically, let a random walk start from $\mathbf{x}_i$, then its probability of first hitting a different data point $\mathbf{x}_j$ before returning to $\mathbf{x}_i$ can be viewed as a measurement of how well connected these two data points are in the weighted graph. Let us introduce the following notation

$$\tau_i^R = \inf\{t \geq 1 \ \mid \ X_t = \mathbf{x}_i\}$$

Then the probability under consideration can be denoted as $P_i(\tau_j < \tau_i^R)$. Use the fact that $P_i(\tau_j < \tau_i^R,\ X_1 = \mathbf{x}_j) = P_{ij}$, and $P_i(\tau_j < \tau_i^R,\ X_1 = \mathbf{x}_i) = 0$, we can compute $P_i(\tau_j < \tau_i^R)$ by the first step analysis:

$$P_i(\tau_j < \tau_i^R) = P_{ij} + \sum_{k\neq i,j} P_{ik}P_k(\tau_j < \tau_i^R) \tag{4.9}$$

Also, for $k \neq i, j$, the following relationship can be established

$$P_k(\tau_j < \tau_i^R) = P_{kj} + \sum_{l\neq i,j} P_{kl}P_l(\tau_j < \tau_i^R) \tag{4.10}$$

Let $V_{i,j}$ be $(P_k(\tau_j < \tau_i^R))_{k\neq i,j}$, and $Q_{i,j}$ be the matrix P with the i,jth rows and columns removed, then (4.10) can be rewritten as

$$V_{i,j} = P(\cdot, j) + Q_{i,j}V_{i,j} \tag{4.11}$$

Where $P(\cdot, j)$ is the jth column of matrix P with the i,jth rows deleted. Thus we can solve for $V_{i,j}$ as the following

$$V_{i,j} = (I - Q_{i,j})^{-1}P(\cdot, j) \tag{4.12}$$

Finally,

$$P_k(\tau_j < \tau_i^R) = P_{ij} + P(i,\cdot)V_{i,j} \tag{4.13}$$

where $P(i,\cdot)$ is the ith row of P with the i,jth columns deleted.

The distance between $\mathbf{x}_i$ and $\mathbf{x}_j$ should be symmetric with respect to these two data points. Thus we can model it as a decreasing function of $\frac{1}{2}(P_i(\tau_j < \tau_i^R) + P_j(\tau_i < \tau_j^R))$. In our experiments, we defined the distance induced by the above first time hitting probability as the following and we call it the absorption distance between $\mathbf{x}_i$ and $\mathbf{x}_j$

$$T(\mathbf{x}_i, \mathbf{x}_j) = [1 - \frac{1}{2}(P_i(\tau_j < \tau_i^R) + P_j(\tau_i < \tau_j^R))]^\beta \tag{4.14}$$

where β is a controlling exponent that can be used to scale the resulting distances.

We applied both the commute distance and the absorption distances in our experiments and derived similar results. They are both proven in these experiments to be efficient in revealing the underlying data structure.

There are of course other types of distances that can be built from the random walks on the graph. For example, for a function $g : \mathbb{R}^p \to \mathbb{R}_+$, the distance defined in the following accumulates the effects of the random walk until it reaches a destination data point for the first time:

$$T_1(\mathbf{x}_i, \mathbf{x}_j) = \frac{1}{2}\left(E_i[\sum_{l=1}^{\tau_j} g(X_l)] + E_j[\sum_{l=1}^{\tau_i} g(X_l)]\right) \tag{4.15}$$

It is clear that the commute distance is a special case of the above distance that can be derived by setting $g \equiv 1$.

We can also make use of the idea in defining the absorption distance, and define the following distance by considering the accumulative effects of the random walk that reach the destination data point before returning back to the starting data point:

$$T_2(\mathbf{x}_i, \mathbf{x}_j) = \frac{1}{2}\left(E_i[\mathbf{1}(\tau_j < \tau_i^R)\sum_{l=1}^{\tau_j} g(X_l)] + E_j[\mathbf{1}(\tau_i < \tau_j^R)\sum_{l=1}^{\tau_i} g(X_l)]\right) \tag{4.16}$$

where $\mathbf{1}(\cdot)$ is the characteristic function.

The commute distance and the absorption distance are both intrinsic to the geometric structure of the dataset, but they clearly have different emphasis. One aspect of their differences lies in the fact that the commute distance accounts for all possible paths linking two data points, including the ones that re-visit other data points and thus take longer time to reach the destination data point. If the given dataset is large, or if the "bandwidth" parameter used in the weight matrix is too big to assign sufficiently small probabilities to these longer paths, the commute distance may be misleading. The absorption distance, on the other hand, only considers the possibilities of reaching the destination data point without re-visiting the starting data point. This, to some extent, can rule out the consideration of some of the longer paths in the calculation.

For the implementation perspective, calculating the commute distance is easier compared to the absorption distance in terms of computational complexity. However, the commute distances tend to have much larger magnitudes because they account for all paths including the longer ones. Thus, as described in the next section, we will have to use a balancing parameter K that has comparable large magnitude in order to balance the penalty term and the summation of weighted commute distances term to be introduced. However, since the absorption distances are always bounded between 0 and 1, the corresponding parameter K is easier to choose when we apply the absorption distances in our novel clustering framework. (See the next section for more details on this.)

4.2 Incorporating the Distance Defined by Random Walks in the FCM Framework with Penalty Term

As described in Chapter 1, fuzzy c-means method generates a fuzzy clustering result by iteratively optimizing an objective function, which is designed to be the total weighted distance between all data points to all centroids. The algorithm updates the choices of centroids and the membership matrix in each iteration such that the objective function decreases. The distance

used in an ordinary FCM algorithm is the Euclidean distance, and the centroids are computed as the average of all data points weighted by the corresponding components in the membership matrix.

The distances defined in the previous section provide better measurements compared to the Euclidean distance when the dataset resides on an unknown manifold which is embedded in a high dimensional space. Thus to incorporate these new distances into the FCM framework may lead to good clustering methods that efficiently cluster datasets with complex structures without dimensionality reduction procedure or knowing the geometric properties of the dataset.

While utilizing the distances defined by random walks, we need to find the centroids of clusters such that it is possible to evaluate a fuzzy clustering result by computing the total weighted distances from data points to centroids. In order to use any distance defined in the previous section, we will use a different interpretation of the word "centroid" by defining it as the representative of a cluster in the sense that every data point in the same cluster can easily reach their centroid by taking random walks on the graph. In this definition, the centroids of clusters are also data points in the given dataset.

Since we expect a good clustering result consists of well separated clusters, the centroids should have large distances between each other. Thus we can modify the FCM framework by penalizing the choices of centroids that are too close together.

Now let the dataset be denoted as $X = \{\mathbf{x}_i\}_{i=1}^N$, and the required number of clusters be denoted as C. Take into consideration of all the above modifications, the novel clustering algorithm, Fuzzy-RW, is designed by performing the following optimization:

$$\min_{U\in\mathfrak{U},\{\mathbf{c}_i\}_{i=1}^C\subset X} \sum_{i=1}^{C}\sum_{j=1}^{N} U_{ij}^2 T^2(\mathbf{x}_j,\mathbf{c}_i) + K\sum_{p\neq q}\frac{1}{T^2(\mathbf{c}_p,\mathbf{c}_q)} \tag{4.17}$$

where $\mathfrak{U} = \{U \in \mathbb{R}^{N\times N} \ : \ 0 \leq U_{ij} \leq 1, \ \sum_{j=1}^N U_{ij} = 1\}$. U is the membership matrix

and U_{ij} measures how likely $\mathbf{x}_j$ is classified into the ith cluster. $T(\cdot,\cdot)$ is one of the distances derived from the random walks on the graph defined in the previous section. For example, $T(\mathbf{x}_i,\mathbf{x}_j)$ can be the commute time distance or the absorption distance between $\mathbf{x}_i$ and $\mathbf{x}_j$. K is a constant used to balance the effects of the penalty term $\sum_{p\neq q}\frac{1}{T^2(\mathbf{c}_p,\mathbf{c}_q)}$ and the summation term in (4.17). Since the commute distances tend to be large, the corresponding clustering algorithm requires a large balancing parameter K. On the other hand, since the absorption distances are bounded by 0 and 1, its corresponding K can be chosen as $O(N)$, where N is the number of data points.

4.3 Utilizing the Local Properties of Datasets in the Weight Matrix

One disadvantage of the objective function in (4.17) that can cause misclassifications is due to the fact that outliers in a dataset have large distances to other data points. Thus, the optimization with the penalization term in (4.17) may prefer outliers as centroids simply because they have large distances to any other data points. This will lead to meaningless clusters, for example, singleton clusters (see Fig.4.2). One way to avoid this problem is to "exaggerate" the distances between outliers and other data points such that the minimization procedure in (4.17) avoids the cases where outliers are chosen as centroids.

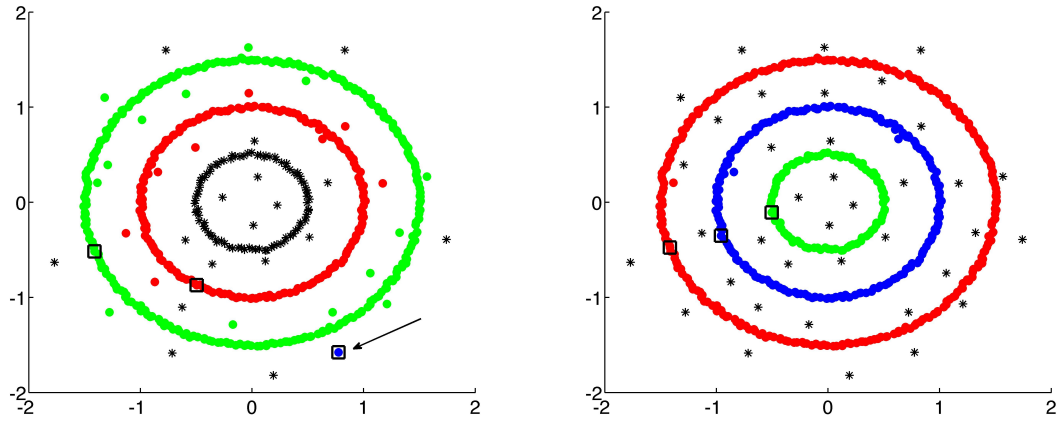
One way to realize this is to modify the weight matrix W using the local density around each data point. That is, for a chosen radius r, we can define the first step neighborhood around $\mathbf{x}_i$ as

$$N_1(\mathbf{x}_i,r)=\{\mathbf{x}_j\in X\ :\ ||\mathbf{x}_j-\mathbf{x}_i||\leq r\} \tag{4.18}$$

where $||\cdot||$ is the Euclidean distance.

Assume the $N_{s-1}(\mathbf{x}_i,r)$ is defined, inductively we can define the s step neighborhood of $\mathbf{x}_i$ $N_s(\mathbf{x}_i,r)$ as the following:

$$N_s(\mathbf{x}_i,r)=\{\mathbf{x}_j\in X\ :\ ||\mathbf{x}_j-\mathbf{x}_k||\leq r,\ \mathbf{x}_k\in N_{s-1}(\mathbf{x}_i,r)\} \tag{4.19}$$



(a) (b)

Figure 4.2: (a) Output of minimizing the objective function ((4.17)) on the data of Fig. 4.1(a). In the absence of information on data density one of the centroids is driven to an outlier datum. (b) Output of the Fuzzy-RW approach incorporated with local density properties (realized by using the weight matrix in (4.20)) when applied to the same dataset. The black squares indicate the locations of the cluster centroids.

The number of elements in the set $N_1(\mathbf{x}_i, r) \bigcup N_2(\mathbf{x}_i, r) \bigcup \cdots \bigcup N_s(\mathbf{x}_i, r)$ can be denoted as $\kappa_{r,s}(\mathbf{x}_i)$. Then for a radius r that is sufficiently small, we can distinguish outliers from data points that belong to the clusters in the given dataset by comparing their κ values, which can be viewed as a measurement for the "density" of data points around the data point under consideration. Thus, for a given r and s we can modify the weight matrix using such density terms:

$$W_{ij} = \exp\left(-\frac{||\mathbf{x}_i - \mathbf{x}_j||^2}{\kappa^{\gamma}(\mathbf{x}_i, \mathbf{x}_j)\sigma}\right) \tag{4.20}$$

where $\kappa(\mathbf{x}_i, \mathbf{x}_j) = \kappa_{r,s}(\mathbf{x}_i)\kappa_{r,s}(\mathbf{x}_j)$ and γ controls the effect of κ in the weight matrix. In the following discussion we may omit the subindex r, s for simplicity.

The weights defined in (4.20) emphasizes the similarities between data points that have higher densities (we expect such data points correspond to those belong to the clusters in the dataset), and the weights on the edges linking outliers to other data points are smaller. Thus the random walks defined using the corresponding modified transition matrix will generate

comparatively larger random walk distance between outliers to other data points.

As an application of the clustering algorithm Fuzzy-RW, we evaluate its performance by demonstrating that Fuzzy-RW outperforms two well-known fuzzy clustering algorithms when applied to the Iris dataset, shown in Fig. 4.3. As is well known, the Iris data se is a benchmark data set commonly employed in pattern recognition analysis Hathaway and Bezdek (2001). It contains three clusters (types of Iris plants: Iris Setosa, Iris Versicolour and Iris Virginica) of 50 data points each in 4 dimensions (features): sepal length, sepal width, petal length and petal width.

The results of applying FCM, spectral method, the bioinformatics-oriented FLAME method Fu and Medico (2007), and Fuzzy-RW to identifying the three clusters embedded in the Iris data set are shown on Table 4.1 and Fig. 4.4. Clearly, in the context of the specific benchmark, Fuzzy-RW outperforms all other three approaches. The parameters involved in each experiment are listed here. FCM method does not require other parameters except the number of clusters is set as 3. The parameters used in the spectral clustering method are $\sigma = 3.03$, $\gamma = 0$, and the first 2 nontrivial eigenvectors of the graph Laplacian were used as the lower dimensional projection of the original dataset, and K-means method was applied on the eigenvectors to generate 3 clusters. Flame method was applied by using the Euclidean distance. 20 nearest neighbors were used to find the "density estimation", 20 nearest neighbors were used to find the "CSO/outlier identification", 20 nearest neighbors were used to find the "Neighborhood approximation"Fu and Medico (2007), and the clusters were formed by posing no specific threshold. For Fuzzy-RW method, parameters are set as $\sigma = 0.1037$, $\gamma = 0$ and $K = 10^{38}$.

4.4 Clustering With Directional Preference

In this section, we consider a type of clustering that prefers clusters consisting data points lining along a given direction. For a user defined directional preference, all data points that line along in different directions should be treated as noise datum in this setting. Here we illustrate this type of clustering with directional preferences using a dataset as in Fig. 4.5 (a), which consists of three lines parallel in a same direction, another line crossing the parallel lines,

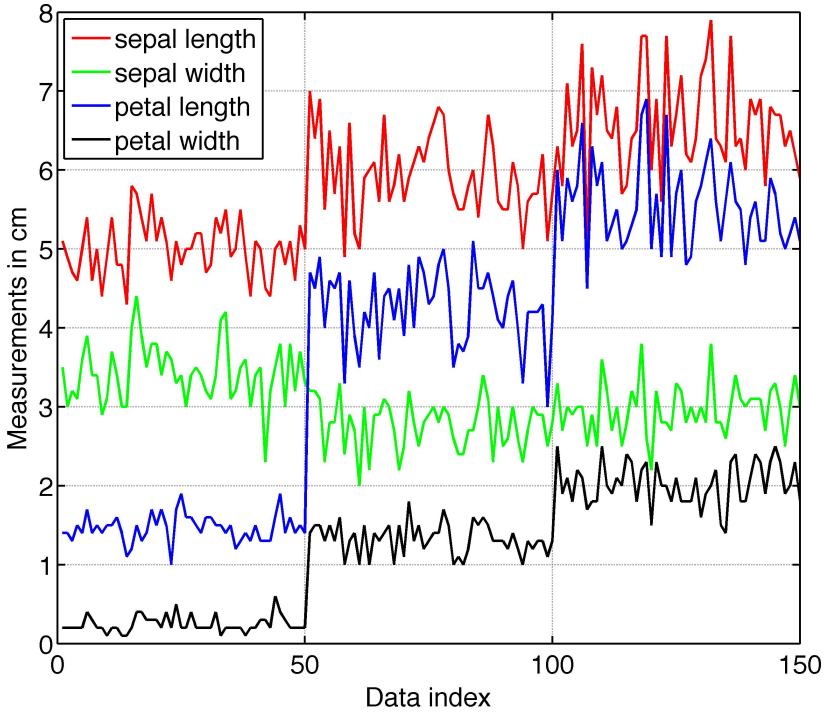


Figure 4.3: The Iris dataset consists of three clusters (each of them a type of Iris plants): Iris Setosa, Iris Versicolour and Iris Virginica. Each cluster contains 50 samples, described by 4 dimensional features: sepal length, sepal width, petal length and petal width.

and noise data points. We will show that by re-defining the weight matrix using the given preferred direction $\mathbf{v}$, we can construct random walks that favor $\mathbf{v}$. Thus the distances derived by using such type of random walks are relatively small for points lining in the direction $\mathbf{v}$, and relatively large for points lining in any other directions.

Let $\mathbf{v}$ be a unit vector indicating the preferred direction. For simplicity, we will now consider $\mathbf{v}$ as a vector in $\mathbb{R}^2$ (the cases when $\mathbf{v} \in \mathbb{R}^n$ can be generalized from this simple case). Then we can denote $\mathbf{v}^\perp$ as the unit vector perpendicular to $\mathbf{v}$ in $\mathbb{R}^2$. Since we would like to collect data points lining in the direction $\mathbf{v}$ together, one way to realize this is to design a new type of distance that shortens the Euclidean distances between such type of data points. Also, the new distances between any other data points lining in a different direction should be accordingly enlarged. Sine $\{\mathbf{v}, \mathbf{v}^\perp\}$ can serve as an orthonormal basis of $\mathbb{R}^2$, we can specify the

	FCM		Spectral		FLAME		Fuzzy-RW	
Index	TP	FP	TP	FP	TP	FP	TP	FP
1	50	0	50	0	50	0	50	0
2	47	13	50	15	50	11	47	2
3	37	3	35	0	37	0	48	3

Table 4.1: The true positive (TP) and false positive (FP) rates obtained by applying FCM, spectral method, FLAME, and Fuzzy-RW respectively. (See text for the parameters used for each algorithm.)

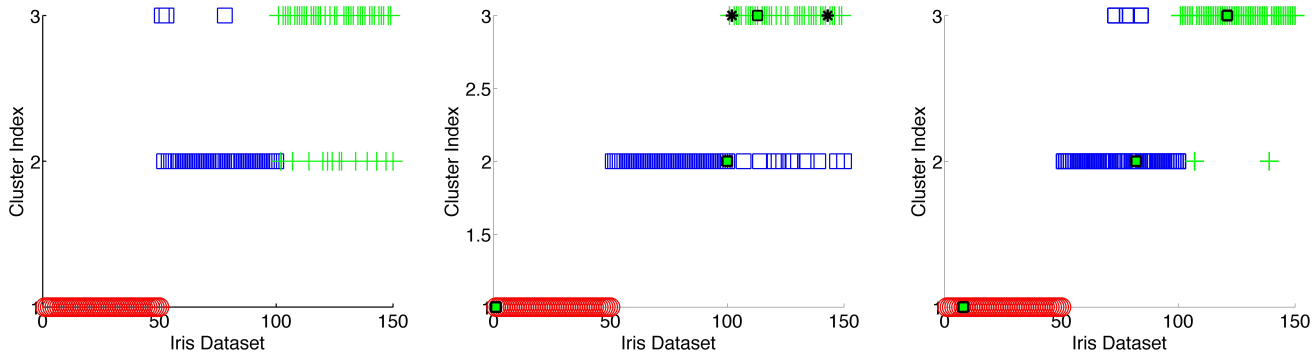


Figure 4.4: (a). Clustering result obtained by applying FCM on the Iris dataset. (b). Clustering result obtained by applying FLAME on the Iris dataset.(c). Clustering result obtained by applying Fuzzy-RW on the Iris dataset. (See text for the details of parameters.)

scale of the "enlarging" and "shortening" with respect to $\mathbf{v}, \mathbf{v}^{\perp}$ as follows.

Let $\hat{d}$ be the new type of distance under development. Suppose $\mathbf{x}_i$ and $\mathbf{x}_j$ are lining in the direction $\mathbf{v}$, and $\mathbf{x}_p$ and $\mathbf{x}_q$ are lining in the direction $\mathbf{v}^{\perp}$. If we would like to "shorten" the distance between $\mathbf{x}_i$ and $\mathbf{x}_j$ by a factor of a ($a \geq 1$) and "enlarge" the distance between $\mathbf{x}_p$ and $\mathbf{x}_q$ by a factor of $1/b$ ($0 < b \leq 1$), then we can write $\hat{d}_{ij} = d_{ij}/a$ and $\hat{d}_{pq} = d_{pq}/b$. Thus we can consider the ratio of the effects of the shortening and enlarging as a/b. Without loss of generality, we can take $b = 1$ and write the ratio of the effects as a ($a \geq 1$). Further, for convenience, we can specify the relationship between the shortening and enlarging using two values between 0 and 1. For example, we can specify the shortening in d_{ij} is by a factor of

$a/(a+1)$ and the enlarging in d_{pq} is by a factor of $a+1$ ($a \geq 1$). Then by adopting the notation of the Mahalanobis distance, we have

$$V = [\frac{a}{a+1}\mathbf{v} \quad \frac{1}{a+1}\mathbf{v}^{\perp}]$$

Let

$$C = VV^T$$

then the new distance $\hat{d}$ between $\mathbf{x}$ and $\mathbf{y}$ is defined as

$$\hat{d}^2(\mathbf{x}, \mathbf{y}) = (\mathbf{x} - \mathbf{y})^T C^{-1} (\mathbf{x} - \mathbf{y})$$

We can use this new distance in defining the weight matrix associated with the random walk on the graph:

$$W_{ij} = \exp\,(-\frac{\hat{d}^2(\mathbf{x}_i, \mathbf{x}_j)}{\kappa^{\gamma}(\mathbf{x}_i, \mathbf{x}_j)\sigma}) \tag{4.21}$$

where the density term $\kappa(\mathbf{x}_i, \mathbf{x}_j)$ is the the same as defined in (4.20) for some given number of step and radius and γ is a parameter that controls the effect of κ in the weight matrix.

In this way, the weights assigned on the edges that link data points lining in the direction of $\mathbf{v}$ (or $-\mathbf{v}$) are enlarged while the weights assigned on edges in other directions are reduced. Thus the corresponding probabilities for taking random walks on the edges are changed accordingly. Since it is more likely for the random walk to proceed in the given direction $\mathbf{v}$, then the expected time to travel between two data points lining in the direction $\mathbf{v}$ is small compared to the ones in any other cases. Using such type of distance in a standard clustering algorithm like FCM or K-means method, we are more likely to obtain clusters that mostly consist of data points lining in the direction $\mathbf{v}$.

Fig.4.5 demonstrates the experimental results of clustering with directional preference. The dataset shown in Fig.4.5 (a) consists of three parallel lines (with perturbations), another line that crosses the parallel lines, and noise data points. Assume that we already know the parallel lines are lining in the direction $\mathbf{v}_0$, then we can define the corresponding Mahalanobis distance and derive the refined random walk, which favors the direction $\mathbf{v}_0$ (or $-\mathbf{v}_0$). Fig. 4.5 (b) shows the clustering result by using the refined distance in FCM framework, with threshold 0.95.

Fir. 4.5 (c) shows the maximum membership values of all data points. In the figures, squares indicate the "centroids" of clusters. The parameters involved are $a = 1.5$, $\sigma = 0.004$, number of steps $s = 4$, radius $r = 0.04$ as in (4.19), weight matrix is defined as in (4.21) with $\gamma = 1$, $K = 10^{38}$ as in (4.17).

4.5 Local PCA Induced Automatic Adaptive Clustering

The random walk induced distance is crucial for the success of the algorithm Fuzzy-RW because it is capable of representing the intrinsic geometric data structures. The "bandwidth" parameter σ used in defining the weight matrix

$$W_{ij} = \exp\left(-\frac{d^2(\mathbf{x}_i, \mathbf{x}_j)}{\kappa^\gamma(\mathbf{x}_i, \mathbf{x}_j)\sigma}\right)$$

serves as a controlling factor of the random walk. Because it is clear to see that for $d^2(\mathbf{x}_i, \mathbf{x}_j)$ smaller than σ, the probability of the random walk to travel between $\mathbf{x}_i, \mathbf{x}_j$ is higher, and vice versa. The induced random walk type of distances are highly affected by this bandwidth parameter, so does the related clustering algorithm. The optimal choice of σ for a given dataset is an active area of research Coifman and Lafon (2006). In this section we describe a method for automatically refining the bandwidth parameter according to the local dataset structures. This type of modification makes the related clustering algorithms generate clusters that have tighter connections between neighboring data points.

To utilize the local neighborhood information around each data point, we first denote the s step neighbors of radius r around data point $\mathbf{x}_i$ as $N_s(\mathbf{x}_i, r)$ as defined in (4.19), $\mathbf{x}_i \in \mathbb{R}^p$. Then by performing PCA on the centered version of the set

$$N(\mathbf{x}_i, r) = N_1(\mathbf{x}_i, r) \bigcup N_2(\mathbf{x}_i, r) \bigcup \cdots \bigcup N_s(\mathbf{x}_i, r)$$

we can find the principal components and the corresponding eigenvalues, denoted by $\{\mathbf{v_k^i}\}_{k=1}^p$ and $\{\lambda_k^i\}_{k=1}^p$, where $\lambda_1^i \geq \lambda_2^i \geq \cdots \lambda_p^i \geq 0$.

The principal components indicate the directions along which the variance of $N(\cdot, r)$ is maximized, and the corresponding eigenvalues can reflect the maximizing effects along different eigenvectors. Then the first few principal components $\{\mathbf{v_k^i}\}_{k=1}^m$ $(m \leq p)$ can serve as a set of basis with each $\mathbf{v}_k^i$ indicating the direction of a coordinate. The origin of such coordinate system is the center of $N(\mathbf{x}_i, r)$, denoted as $\mathbf{x}_i^*$. Then in order to generate clusters where neighboring data points are "tied" closely together, we would like to shorten the distances in the first few principal directions $\{\mathbf{v_k^i}\}_{k=1}^m$ $(m \leq p)$ by factors $\{\lambda_k^i\}_{k=1}^m$ respectively. This can be done by first centering all data points such that $\mathbf{x}_i^*$ is the origin in the new coordinate system, then shortening the distances along each $\mathbf{v_k^i}$ with the factor λ_k^i, $k = 1, 2, 3, \cdots, m$. Without loss of generality, we can use $\frac{\lambda_k^i}{\sum_{k=1}^m \lambda_k^i}$ as the shortening factors. The method of shortening is described in the previous section, here we describe this process using the simple procedure performed in $\mathbb{R}^2$ with $m = p = 2$ (the cases of $\mathbb{R}^n$ and/or $m < p$ can be generalized from here).

For a fixed data point $\mathbf{x}_i \in \mathbb{R}^2$ in the given dataset $X = \{\mathbf{x}_k\}_{k=1}^N$, a given number of step s and radius r, we can write $N_s(\mathbf{x}_i, r)$ as the set of s step neighbors of $\mathbf{x}_i$. Then let $\{\mathbf{v}_1^i, \mathbf{v}_2^i\}$ be the two orthonormal principal components of the set $D = \{\mathbf{y} - \mathbf{x}_i^* \mid y \in N(\mathbf{x}_i, r)\}$ with associated eigenvalues are λ_1^i, λ_2^i, where $\mathbf{x}_i^*$ is the center of $N(\mathbf{x}_i, r)$ and $N(\mathbf{x}_i, r) = N_1(\mathbf{x}_i, r) \bigcup N_2(\mathbf{x}_i, r) \bigcup \cdots \bigcup N_s(\mathbf{x}_i, r)$. For simplicity, here we use the notation $\mathbf{v}_1 = \mathbf{v}_1^i$, $\mathbf{v}_2 = \mathbf{v}_2^i$, $\lambda_1 = \lambda_1^i$ and $\lambda_2 = \lambda_2^i$. To set up the new coordinate system, we can shift the original dataset X to $\tilde{X} = \{\tilde{\mathbf{x}_k} \mid \tilde{\mathbf{x}_k} = \mathbf{x}_k - \mathbf{x}_i^* \ , k = 1, 2, \cdots, N\}$. Let $\hat{\lambda}_i$ be $\frac{\lambda_i}{\lambda_1 + \lambda_2}$, $i = 1, 2$, using the notation introduced in the previous section, we can define

$$V = [\hat{\lambda_1}\mathbf{v}_1 \ \ \hat{\lambda_2}\mathbf{v}_2] \tag{4.22}$$

and $C = VV^T$. Then the Mahalanobis distance $\hat{d}$ between $\mathbf{x}_i$ (in X) and any data point $\mathbf{x}_j$ (in X) is defined through the following:

$$\hat{d}^2(\mathbf{x}_i, \mathbf{x}_j) = (\tilde{\mathbf{x}_i} - \tilde{\mathbf{x}_j})^T C^{-1} (\tilde{\mathbf{x}_i} - \tilde{\mathbf{x}_j})$$

This modified distance can be used in defining the weight matrix

$$W_{ij} = \exp\left(-\frac{\hat{d}^2(\mathbf{x}_i, \mathbf{x}_j)}{\kappa^\gamma(\mathbf{x}_i, \mathbf{x}_j)\sigma}\right)$$

where γ is a parameter that decides the effect of the density term in the weight matrix. Eventually, the above new weight matrix induces the random walk type of distances, which can be used in a standard clustering algorithm. Here, we still have to choose the "bandwidth" parameter σ, but the effect of σ can be modified by the shortening process represented in $\hat{d}$. Thus, the "bandwidth" is not a parameter that affects all data points uniformly. Instead, it is automatically adjustable through the above process.

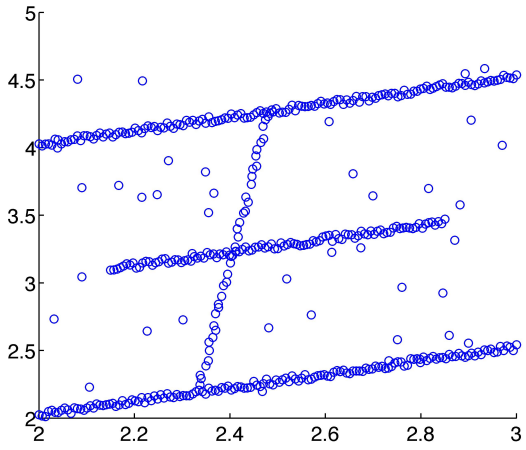
Regularization on the above shortening procedure can be introduced to avoid extreme cases. We still use the dataset $X \in \mathbb{R}^2$ as an example. To avoid the cases where $\hat{\lambda_1}$ is too large compared to $\hat{\lambda_2}$ which makes the above shortening procedure too restricted in the first principal component $\mathbf{v}_1$, we can introduce a controlling constant c when we define V in (4.22):

$$\hat{\lambda_1} = \begin{cases} \frac{\lambda_1}{\lambda_1+\lambda_2} & \text{if } \frac{\lambda_1}{\lambda_2} \leq c \\ \frac{c}{c+1} & \text{if } \frac{\lambda_1}{\lambda_2} > c \end{cases} \tag{4.23}$$

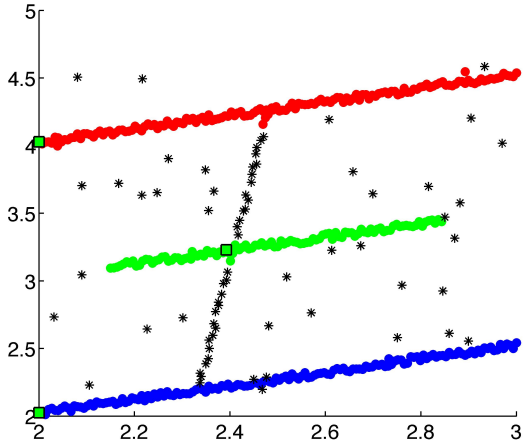
$\hat{\lambda_2}$ can be computed as $\hat{\lambda_2} = 1 - \hat{\lambda_1}$ in $\mathbb{R}^2$.

Here we take a dataset shown in Fig. 4.6. Fig. 4.6 (a) shows the clustering result by using Fuzzy-RW with commute distance and (b) shows the maximum membership value at each data point. Fig. 4.7 (a) shows the clustering result by using Fuzzy-RW and commute distance, incorporated with the local PCA and automatic adjustment described in this section. The related parameters are $\sigma = 0.017$, $\gamma = 0$, $K = 10^{25}$ and the threshold is set to be 0.7. Fig.4.7 (b) shows the maximum membership value at each data point. The related parameters are $s = 2$, $r = 0.06$, $a = 1.5$, $\sigma = 0.017$, $\gamma = 0$, $K = 10^{36}$, and the threshold is set to be 0.95.

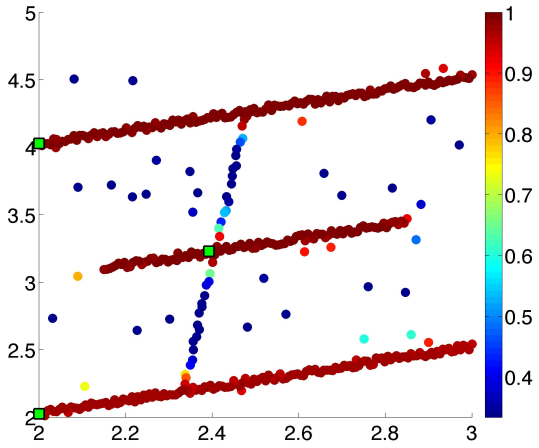
It is clear from the figures that the local PCA adjustment makes the clustering algorithm generate clusters that have tight local connections. For some dataset whose clusters consist of data points scattering in the form of line segments, this type of adjustment is especially useful for detecting the segments.



(a)

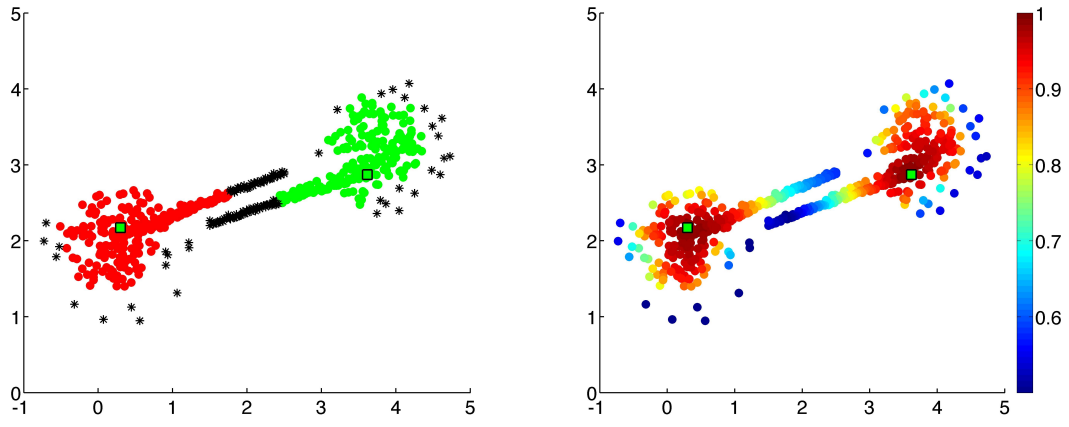


(b)



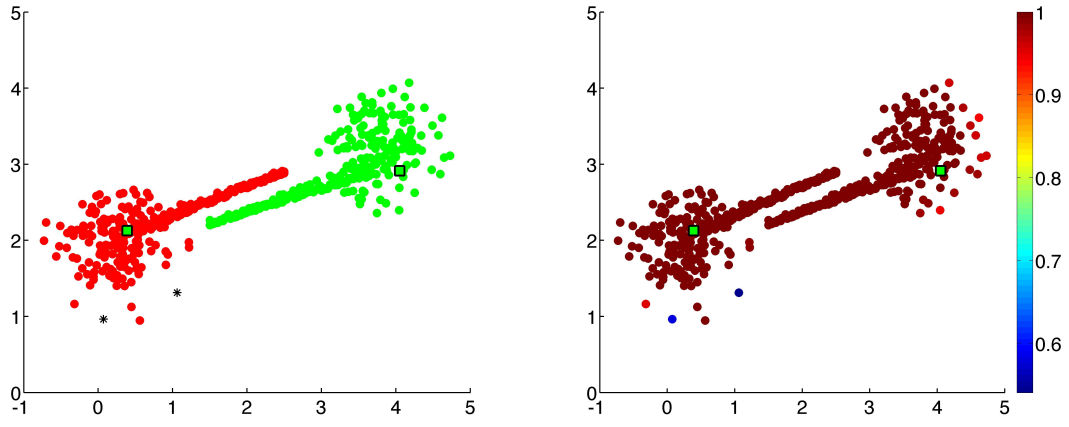
(c)

Figure 4.5: (a). A dataset perturbed by noise datum. This dataset is used to demonstrate the technique of clustering with directional preference. (b). The clustering result obtained by specifying a directional preference and posing the threshold as 0.75 (see text for details). (c). The maximum membership values at each data point.



(a) (b)

Figure 4.6: (a). Clustering result derived by using Fuzzy-RW. Threshold is set to be 0.7. (b). Maximum membership values at each data point. (See text for the parameters involved.)



(a) (b)

Figure 4.7: (a). Clustering result derived by using Fuzzy-RW incorporated with local PCA. Threshold is set to be 0.95. (b). Maximum membership values at each data point. (See text for the parameters involved.)

CHAPTER 5. Face Recognition

5.1 Face Recognition by Using Lower Dimensional Linear Subspaces

The designs of face recognition algorithms are complicated due to the variability of a same person's appearance viewed under various lighting conditions and (or) different postures. Two general methodologies that have been proposed and used in applications are either to compare images based on some properties that are insensitive to the variability caused by the imaging conditions, or to model the variability of the images of each individual.

Approaches developed with the first type of methodology mainly focus on comparing images of an object using edges in the images, which are the discontinuities in the image intensity and are caused by the discontinuities of the albedo on the surface and the boundary of the object. Although the edges detected in images tend to be insensitive to a range of illumination conditions Binford (1988), they do not offer all useful information for face recognition by only providing the discontinuities of the image intensity. Also, only a small fraction of the edges are shared by images taken under a broad range of lighting conditions Belhumeur and Kriegman (1996), and it has been observed that the variability of edges caused by different illumination conditions are often greater than that caused by the change of individual Adini et al. (1997).

Although the variations of the image intensity are difficult to trace under changing lighting conditions, the "appearance-based" approaches have been developed in order to take into account of more useful information and to overcome the above drawbacks of the approaches based only on image edges. The principle of such appearance-based approaches is to model the image variations caused by changing illumination conditions and/or postures by extracting

representative information from training sets. However, one of the drawbacks of such methods lies in the fact that in order to achieve satisfactory recognition results for an individual, the lighting condition and/or posture in the testing image has to appear in the training set for this individual. The training set will be unnecessarily large if one attempts to include all possible illumination conditions and/or postures for all individuals. Thus, more generic approaches of modeling the images with variations under different lighting conditions and/or postures are necessary for face recognition.

One generic approach that focuses on the shape and properties of the whole set of images of an individual taken under changing conditions is proposed in Belhumeur and Kriegman (1996) and further developed and optimized using different techniques and/or focusing on different perspectives in Basri and Jacobs (2003), chih Lee et al. (2005), Georghiades et al. (2000), etc. It is shown in Belhumeur and Kriegman (1996) that under the illumination of arbitrary number of point light sources at infinity, the set of n-pixel monochrome images of a convex object with a Lambertian reflectance function forms a convex polyhedral cone in $\mathbb{R}^n$, which is called the illumination cone. Also, the dimension of this illumination cone equals the number of distinct surface normals. Further, the set of n-pixel images of an object of any shape with a more general reflectance function, illuminated under all lighting conditions, also forms a convex cone in $\mathbb{R}^n$. These results suggest the recognition approach that first models the illumination cone for every individual, then classifies a testing image to an individual in the training set according to the image's nearest illumination cone. This approach should be able to achieve sufficient results if the illumination cones are well separated.

This chapter is organized as follows. Section 1 reviews lower dimensional linear subspace representations of the illumination cone. Section 2 discusses several popular dimensionality reduction methods. Section 3 focuses on the description of applying Fuzzy-RW in solving face recognition problems.

5.1.1 Approximation of the Illumination Cones by Harmonic Basis

Based on the results given in Belhumeur and Kriegman (1996) on the illumination cones, object recognition algorithms have been developed Georghiades et al. (1998) Georghiades et al. (2000) using the projections of the illumination cones to lower dimensional subspaces for simplicity and efficiency. There are also experimental results shown in Hallinan (1994) Epstein et al. (1995) Yuille et al. (1999) which demonstrate the fact that large numbers of images of real, Lambertian objects taken under changing lighting conditions lie near a lower dimensional linear space. With error tolerance, these experimental results suggest using such lower dimensional space as an approximation of the illumination cones in object recognitions. The authors of Basri and Jacobs (2003) proposed a 9 dimensional linear space as an approximation of the illumination cone using spherical harmonic functions to model the lighting conditions and representing the Lambertian reflections as an analog of a convolution. Such a convolution turns out to be a low pass filter, which supports their results on the lower dimensional approximation of the illumination cone.

The objects under various illuminations considered in Basri and Jacobs (2003) are convex ones in order to avoid discussions of cast shadows, which occur when parts of the object are blocking the light from reaching other parts of the object. Thus the images of these objects only have attached shadows, which occur when the light source move away from the viewing direction. The light sources under consideration are the "distant" ones such that every point on the object is illuminated by the lighting coming from a same direction. Also, the surface of the objects are assumed to reflect light according to Lambert's law, which indicates that the surface materials absorb and reflect light uniformly in all directions. The only parameter in their model is albedo at each point on the surface, which determines the fraction of the light to be reflected. The discussion of the set of all possible images produced under various lighting conditions is broken down into two steps. The first step describes the reflection function, which represents the amount of light that is reflected by each surface normal under different illumination conditions. It depends only on the lighting conditions and Lambertian reflectance and

it is independent of the object structure. The second step describes the transform between the reflection function and the images. This step depends only on the object structure and albedo.

First, the intensity of light can be viewed as a function of its direction and the reflection can be described as a function of surface normal. Thus, these two functions are functions defined on the surface of the unit sphere S^2 centered at the origin. Let u_l and v_r be the unit vectors related to the lighting and reflection directions. Then the light coming from direction u_l with intensity l can be represented as $l(u_l)$. According to the Lambert's law, when the light described by $l(u_l)$ reaches at a surface point with normal direction v_r and albedo ρ, the intensity reflected is given by

$$i = l(u_l)\rho \max(u_l \cdot v_r, 0) \tag{5.1}$$

For a fixed lighting, if we ignore the albedo, the above reflected intensity is a function of v_r. If we write $k(u \cdot v) = \max(u \cdot v, 0)$ and assume the light comes from multiple directions, then the total reflected light $r(v_r)$ can be written as

$$r(v_r) = \int_{S^2} k(u_l \cdot v_r) l(u_l) du_l \tag{5.2}$$

The above expression is analogous to a convolution, and we can define the following notation

$$r(v_r) = k * l := \int_{S^2} k(u_l \cdot v_r) l(u_l) du_l \tag{5.3}$$

Before we derive the properties of the above convolution, it is necessary to include some discussions of the spherical harmonics and the Funk-Hecke Theorem.

The surface spherical harmonics are convenient tools for performing convolutions on the sphere. They are a set of functions that form an orthonormal basis for all functions defined on the surface of the sphere. They are denoted as Y_{nm} with $n = 0, 1, 2, \cdots$, $m = -n, -n+1, \cdots, n-1, n$. (It is sometimes convenient to represent the unit vector u using a pair of angles (θ, ϕ), where $u = (x, y, z) = (\cos\phi \sin\theta, \sin\phi \sin\theta, \cos\theta)$.)

$$Y_{nm}(\theta, \phi) = \sqrt{\frac{(2n+1)(n-|m|)!}{4\pi(n+|m|)!}} P_{n|m|}(\cos\theta) e^{im\phi} \tag{5.4}$$

where P_{nm} are the associated Legendre functions defined as

$$P_{nm}(z) = \frac{(1-z^2)^{m/2}}{2^n n!}\frac{d^{n+m}}{dz^{n+m}}(z^2-1)^n \tag{5.5}$$

Y_{nm} is called an nth order harmonics.

The first nine harmonics written in x, y, z are:

$$Y_{00} = \frac{1}{\sqrt{4\pi}}, \;\; Y_{10} = \sqrt{\frac{3}{4\pi}}z$$

$$Y_{11}^e = \sqrt{\frac{3}{4\pi}}x, \;\; Y_{11}^o = \sqrt{\frac{3}{4\pi}}y$$

$$Y_{20} = \frac{1}{2}\sqrt{\frac{5}{4\pi}}(3z^2-1), \;\; Y_{21}^e = 3\sqrt{\frac{5}{12\pi}}xz$$

$$Y_{21}^o = 3\sqrt{\frac{5}{12\pi}}yz, \;\; Y_{22}^e = \frac{3}{2}\sqrt{\frac{5}{12\pi}}(x^2-y^2)$$

$$Y_{22}^o = 3\sqrt{\frac{5}{12\pi}}xy$$

where e and o represent the even and odd components of the harmonics respectively as in $Y_{nm} = Y_{n|m|}^e \pm iY_{n|m|}^o$.

Any piecewise continuous function f defined on the surface of the sphere can be written as an infinite series of harmonics:

$$f(u) = \sum_{n=0}^{\infty}\sum_{m=-n}^{m=n} f_{nm}Y_{nm}(u) \tag{5.6}$$

where the coefficients f_{nm} are computed by:

$$f_{nm} = \int_{S^2} f(u)Y_{nm}^*(u)du \tag{5.7}$$

where $Y_{nm}^*(u)$ is the complex conjugate of Y_{nm}.

Thus, the lighting function l can be represented by the following:

$$l = \sum_{n=0}^{\infty}\sum_{m=-n}^{m=n} l_{nm}Y_{nm} \tag{5.8}$$

Because of the circularly symmetric property of the Lambertian kernel, it can be shown that

$$\int_{S^2} k(u) Y_{nm}^*(u) du = 0, \;\; m \neq 0 \tag{5.9}$$

Thus the kernel k can be written as:

$$k = \sum_{n=0}^{\infty} k_n Y_{n0} \tag{5.10}$$

Then by using the following Funk-Hecke Theorem stated specifically for the current problem setting, we can derive the reflection function using spherical harmonics.

Theorem 19 (Funk-Hecke Theorem)**.** *Let $k(u \cdot v)$ be a bounded, integrable function on $[-1, 1]$. Then*

$$k * Y_{nm} = \alpha_n Y_{nm}$$

where

$$\alpha = \sqrt{\frac{4\pi}{2n+1}} k_n$$

Thus, the reflection function r, being a convolution of Lambertian kernel k and the lighting function l, can be written as:

$$r = k * l = \sum_{n=0}^{\infty} \sum_{m=-n}^{m=n} (\alpha_n l_{nm}) Y_{nm} \tag{5.11}$$

From the above representation, it is clear that after the convolution of l with the kernel k, every amplitude l_{nm} is scaled by a factor α_n that depends only on the kernel k. Treating l as a signal and k as a filter, we can later see how the amplitude of l changes after passing through the filter.

Through computations detailed in Basri and Jacobs (2003), the coefficients in the harmonic expansion of k can be derived as:

$$k_n = \begin{cases} \frac{\sqrt{\pi}}{2} & n = 0 \\ \sqrt{\frac{\pi}{3}} & n = 1 \\ (-1)^{n/2+1} \frac{\sqrt{(2n+1)\pi}}{2^n (n-1)(n+2)} C_n^{n/2} & n \geq 2, \text{even} \\ 0 & n \geq 2, \text{odd} \end{cases} \tag{5.12}$$

And the first few coefficients are evaluated as the following:

$$k_0 = \frac{\sqrt{\pi}}{2} \approx 0.8862, \quad k_1 = \sqrt{\frac{\pi}{3}} \approx 1.0233$$

$$k_2 = \frac{\sqrt{5\pi}}{8}, \quad k_4 = -\frac{\sqrt{\pi}}{16} \approx -0.1108$$

$$k_6 = \frac{\sqrt{13\pi}}{128} \approx 0.0499, \quad k_8 = \frac{\sqrt{17\pi}}{256} \approx -0.0285$$

while $k_3 = k_5 = k_7 = 0$, and $|k_n|$ approaches zeros as $O(n^{-2})$.

The square of the coefficients divided by the total squared energy in the harmonic expansion is often used as the measure of the energy captured in each respective harmonic term. If we write the kernel function in terms of angles θ, ϕ, the total energy can be computed as the following:

$$\int_0^{2\pi} \int_0^{\pi} k^2(\theta) \sin\theta d\theta d\phi = 2\pi \int_0^{\pi/2} \cos^2\theta \sin\theta d\theta = \frac{2\pi}{3} \tag{5.13}$$

It can be seen from the accumulated sum of the first few energy terms that a second order approximation accounts for 99.22% of the total energy:

$$(\frac{\pi}{4} + \frac{\pi}{3} + \frac{5\pi}{64})/\frac{2\pi}{3} \approx 99.22\%$$

Such a second order approximation of k can be written as

$$k(\theta) \approx \frac{3}{23} + \frac{1}{2}\cos\theta + \frac{15}{32}\cos^2\theta$$

Thus the Lambertian kernel k can be considered as a low pass filter, which means that the high frequency components in the light function l will not contribute much to the convolution between k and l. We can achieve a lower dimensional approximation of the reflectance function for a choice of N

$$r = k * l \approx \sum_{n=0}^{N} \sum_{m=-n}^{m=n} (\alpha_n l_{nm}) Y_{nm} \tag{5.14}$$

For every order n, there are $2n + 1$ harmonics involved. The first order approximation involves 4 harmonics, the second order approximation involves 9 harmonics, and the third order approximation involves 18 harmonics. According to Basri and Jacobs (2003), better approximation accuracy are expected if the light sources includes enhanced diffuse components of

low frequency and worse results are anticipated when the light consists of mainly high frequency patterns. A lower bound for the approximation accuracy is computed in Basri and Jacobs (2003) and it is shown that the accuracy of a second order approximation for any light function exceeds 97.96%. And with a fourth order approximation, the accuracy exceeds 99.48%.

For convenience, we can write r_{nm} as the reflectance produced by the basis vector Y_{nm}, and call it the harmonic reflectance:

$$r_{nm} = k * Y_{nm} = \alpha_n Y_{nm} \tag{5.15}$$

Then the reflectance function can be written as

$$r = k * l \approx \sum_{n=0}^{N} \sum_{m=-n}^{m=n} l_{nm} r_{nm} \tag{5.16}$$

After deriving the formula for the reflectance function, the transformation from r to the related image is made in the following way: each point of the object inherits its intensity from the point on the sphere whose normal direction is the same. Further, the intensity is scaled by its albedo.

To be specific, let p_i be the i-th object point, n_i be the surface normal at p_i, and ρ_i the albedo of p_i. Let the reflectance function be $r(n_i)$, then the intensity at p_i can be expressed as:

$$I_i = \rho_i r(n_i) = \rho_i \sum_{n=1}^{\infty} \sum_{m=-n}^{m=n} l_{nm} r_{nm}(n_i) \tag{5.17}$$

Then, the related image can be written as the following

$$I_i = \sum_{n=0}^{\infty} \sum_{m=-n}^{m=n} l_{nm} b_{nm}(p_i) \tag{5.18}$$

where b_{nm} are called harmonic images:

$$b_{nm}(p_i) = \rho_i r_{nm}(n_i) \tag{5.19}$$

The transformation between the reflectance and the related image may affect the representation results. The discussion made in Basri and Jacobs (2003) indicates that, although such

effects may make the results arbitrarily bad, in typical cases, the approximation results will not be less accurate. Also, the above models for lighting conditions, reflectance function, and the transformation between the reflectance and the related image made in Basri and Jacobs (2003) still ignore some real world effects including the surface being deviated from Lambertian surface, the object being non-convex, the effects caused by cast shadows, and effects caused by noise.

After deriving the above models for the images taken under changing lighting conditions, a face recognition algorithm based on such analytic models can be constructed. The experiments done in Basri and Jacobs (2003) assume that the individuals are facing the camera, and the database consists of information of their surface normals and albedos. The recognition process is to first build the models for each individual, then compute the distances from a testing image to these models, and finally classify the testing image to the model that has the minimum distance to it. Here we omit the exact procedure done in their experiments, but only mention that accurate results are obtained even with $4D$ linear approximations. Also, compared to other face recognition algorithm, for example, the ones where PCA is performed to derive a lower dimensional linear approximation from a set of samples of individuals' images, the proposed algorithm uses analytic descriptions of the lower dimensional approximation. Such analytic description has at least two advantagesBasri and Jacobs (2003): firstly, such approximation provides the accurate error estimation, unlike other approximations whose errors vary by samples of iamges; secondly, such approximation provides better efficiency compared to PCA approaches, and efficiency is greatly appreciated when the algorithm needs to be done on the run.

5.1.2 Acquiring Subspaces Under Variable Lighting Conditions For Face Recognition

Since the whole set of images of individuals taken under different lighting conditions form a convex cone (illumination cone), different face recognition algorithms have be developed

attempting to approximate this set using lower dimensional subspaces. A typical type of approaches relies on performing PCA on a sample set of images of individuals, then uses the principal vectors as the basis of the lower dimensional linear subspace. The authors of chih Lee et al. (2005) derived the conclusion on how to arrange the lighting conditions for the training set, such that the corresponding images can be directly used as the basis vectors of the lower dimensional linear subspace approximating the illumination cone.

The authors of chih Lee et al. (2005) develop their approximation technique based on the results of using a 9 dimensional linear subspace as an approximation of a illumination cone, which is given in Basri and Jacobs (2003). That is, this result is interpreted as the proof of the following statement: for all individuals there exist nine universal virtual lighting conditions such that the images taken under these lighting conditions can be used to approximate the illumination cones, and these images are called harmonic images in Basri and Jacobs (2003). Since these harmonic images are sometimes not real images (some pixels have negative values), they must be derived from computations of real images or rendered from some geometric models of individuals under synthetic harmonic lighting conditions, which require the information of the surface normals and albedo. Such constructions are usually physically difficult to realize. Thus the authors of chih Lee et al. (2005) attempt to construct a linear subspace R as a good approximation of the 9 dimensional subspace H using lighting conditions that can be easily realized.

In their discussion, only the single distant isotropic light sources are considered. A finite sized subset Ω of the unit sphere S^2 is associated with the set of light sources. Ω that is considered in the discussion will be either uniformly sampled on the sphere or the hemisphere. Given Ω and an integer d, a subset of directions $\{s_1, s_2, \cdots, s_d\}$ and the associated lighting directions $\{l_{s_1}, l_{s_2}, \cdots, l_{s_d}\}$ give rise to d images that can be used to approximation the illumination cone C and/or H. Two algorithms are proposed in chih Lee et al. (2005), the first one searches for such approximation by minimizing a type of distance between H and R, and the second one incorporates the maximization of the volume of $R \bigcap C$ on top of the minimization in the first

algorithm. Moreover, it turns out that such optimizations lead to a universal set of lighting conditions under which the images taken can be used as basis vectors of R for all individuals. And in some cases, as few as five training images for each individual can produce accurate recognition results if small error can be tolerated.

The distance between two spaces H and R can be measured by the sum of squared cosines of the principal angles between them. Let the orthonormal columns of matrices A and B represent the basis vectors of R and H, then the distance between R and H can be computed using the singular values of $B^T A$. That is, if we denote the singular values as $\{\alpha_1, \cdots, \alpha_k\}$, where k is the minimum of the dimensions of R and H, then the similarity between R and H is:

$$Sim(R, H) = \sum_{i=1}^{k} \alpha_i^2 \tag{5.20}$$

To search for the optimal subspace R that maximize the above similarity from all possibilities is a difficult task and therefore a local greedy algorithm is needed for reasonable approximation. It is proposed in chih Lee et al. (2005) that a sequence of nested linear subspaces $R_0 \subset R_1 \subset \cdots \subset R_9 = R$ can be computed to derive R by searching for "extreme ray" x_i, $i \geq 0$. Let R_0 be the empty set, Ω_i be the set obtained by deleting the i extreme rays from Ω, and $\Omega_0 = \Omega$. Then R_i and Ω_i can be computed inductively from the known R_{i-1} and Ω_{i-1}:

$$R_i = R_{i-1} \oplus x_i, \quad \Omega_i = \Omega_{i-1} \setminus x_i \tag{5.21}$$

where

$$x_i = \arg \max_{x \in \Omega_{i-1}} Sim(x \oplus R_{i-1}, H) \tag{5.22}$$

Some preliminary experiments reveal that the configuration that such nested optimization generates has some special properties. First, we can denote the lighting directions in terms of (θ, ϕ), where ϕ is the elevation angle of range $0 \leq \phi \leq 180^\circ$ and θ is the azimuth angle of range $-180^\circ \leq \theta \leq 180^\circ$. Then the optimized configuration is found consisting of 2 frontal directions (directions with small ϕ values), 5 side directions ($\phi \approx 90^\circ$) with θ values spread quasi-uniformly around the lateral rim of the unit sphere, 1 direction with $\phi > 90^\circ$ and the last direction that appears being chosen at random. The important result is, across all individuals

in their experiment, the configurations are very similar. More detailed experiments and direct computations are performed in chih Lee et al. (2005) on a set of images from the Yale Database to derive the optimal configuration of the lighting directions. The direct computations indicate a universal configuration across all 10 individuals.

A second algorithm is proposed in chih Lee et al. (2005) to overcome the drawback of the above algorithm as it requires long computation time. Also, it attempts to describe the geometric relationship between R and the illumination cone C more specifically. Thus, this algorithm aims to generate an approximation R such that the distance between R and H is minimized and the volume $vol(C \cap R \cap B_1)$ is maximized (B_1 is the unit ball). It is proved in chih Lee et al. (2005) that the following can be taken as a good approximation of $C \cap R$:

$$R_C = \left\{ x \mid x \in R, x = \sum_{i=1}^{k} \alpha_i x_i, \;\; \alpha_i \geq 0 \right\} \tag{5.23}$$

where $x_i \in \Omega$ is a basis of R.

Then, the computation of R can be formulated inductively. Let Ω_i be the set obtained by deleting i extreme rays from Ω, then R_i is the space spanned by x_i and R_{i-1}, and Ω_i is $\Omega_{i-1} \setminus x_i$, where

$$x_i = \arg \max_{x \in \Omega_{i-1}} \frac{dist(x, R_{i-1})}{dist(x, H)} \tag{5.24}$$

$R_0 =$ and $dist(x, R_0)$ is defined to be 1. And the distance between x_i and H or R_i is defined to be the L_2 distance between a point and a subspace. The related experiments show since neither a singular value decomposition nor a Gramm-Schmidt process are computed, this algorithm runs two to three times faster than the first algorithm. Also, the general characteristic of the configurations of lighting directions obtained here are similar to the ones derived before.

How to find a fixed configuration of the nine lighting directions across all individuals is further investigated in chih Lee et al. (2005). To solve this problem, the optimization procedure of the previous algorithm is modified by computing the average of the quotient in (5.24). That is, a nested linear subspaces $R_0 \subset R_1 \subset \cdots \subset R_i \subset \cdots \subset R_9 = R$ can be computed by selecting

x_i such that

$$x_i = \arg \max_{x \in \Omega_{i-1}} \sum_{k=1}^{l} \frac{dist(x^k, R_{i-1}^k)}{dist(x^k, H^k)} \tag{5.25}$$

where x^k is the image of the kth individual taken under the light direction x, H^k is the harmonic subspace that can be constructed for the kth individual, and R_{i-1}^k is the linear subspace spanned by the images $\{x_1^k, \cdots, x_{i-1}^k\}$ taken under the $i-1$ general lighting directions $\{x_1, \cdots, x_{i-1}\}$. The results of general configurations are computed as the following

$$\{(0,0), (68,-90), (74,108), (80,52), (85,-42), (85,-137), (85,146), (85,-4), (51,67)\}$$

The experimental face recognition on the Yale Database B using the above configuration of lighting directions achieves good results chih Lee et al. (2005). To be specific, it always performs better than a randomly generated configuration of lighting directions. Also, recognition experiments using the sequence of nested subspaces indicate that subspaces with dimension greater than four in this sequence all give remarkably good results. Comparisons between this algorithm and other types of face recognition algorithms including Eigenfaces, Nearest Neighbor method, and recognition using harmonic images are made in chih Lee et al. (2005). It is shown that this proposed algorithm outperforms the other types of face recognition methods, and it does not need any training set.

5.2 Appearance-Based Face Recognition

Appearance-based algorithms account for a large part of face recognition algorithms. These methods attempt to find lower dimensional approximations of the set of images taken under different lighting conditions and poses, and such approximations should be invariant or insensitive to the variances caused by lighting conditions or postures. Below we discuss three types of popular appearance-based face recognition methods. They are recognition by Eigenfaces Turk and Pentland (1991b) Turk and Pentland (1991a) Belhumeur et al. (1997) , Fisherfaces Belhumeur et al. (1997), and Laplacianfaces He et al. (2005), respectively. These methods all make use of training sets in order to search for the basis vectors of corresponding linear lower dimensional subspaces that approximate the sets of all images. Recognition by Eigenfaces finds such approximation by maximizing the total covariance of the training set, Fisherfaces are

found as the basis vectors that maximize the inter-class covariance and minimize the intra class covariances, and Laplacianfaces are found as those that are able to capture the local structures of the image manifold. It is further proved in He et al. (2005) that in fact, Eigenfaces, Fisherfaces and Laplacianfaces can be derived from different choices of graph models, with different focuses. In each section below, techniques of Eigenfaces, Fisherfaces and Laplacianfaces are described respectively.

5.2.1 Face Recognition by Eigenfaces

For a given training set $\{\mathbf{x}_1, \cdots, \mathbf{x}_N\} \subset \mathbb{R}^n$, if we target at finding a lower dimensional linear subspace that approximate the whole training set, and assume such procedure is built upon a linear transformation of the images in the training set, then such transformation can be written as

$$\mathbf{y_i} = W^T \mathbf{x}_i, \quad i = 1, 2, \cdots, N \tag{5.26}$$

where $W \in \mathbb{R}^{n \times m}$ is the transformation matrix with orthonormal columns.

The total scatter matrix S_T of the original training set is defined as

$$S_T = \sum_{k=1}^{N} (\mathbf{x} - \mu)(\mathbf{x} - \mu)^T \tag{5.27}$$

where $\mu \in \mathbb{R}^n$ is the mean image of the training set. Then the total scatter of the transformed images is

$$S_T^y = \sum_{k=1}^{N} W^T (\mathbf{x} - \mu)(\mathbf{x} - \mu)^T W = W S_T W \tag{5.28}$$

By performing PCA, the optimal transformation W_{opt} is chosen to be the one that maximizes the determinant of the transformed scatter S_T^y:

$$W_{opt} = \arg\max_W |W^T S_T W| \tag{5.29}$$

If we choose to find an m-dimensional linear space to approximate the training set, then the transformation W_{opt} is an $n \times m$ matrix, $W_{opt} = [\mathbf{w}_1 \mathbf{w}_2 \cdots \mathbf{w}_m]$. Each column $\mathbf{w}_i$ is the eigenvector corresponding to the i-th largest eigenvalue.

After deriving the transformation W_{opt}, whose columns are the basis vectors of the subspace that maximizes the total variance, every testing image $\mathbf{x}$ can be projected onto this subspace:

$$\mathbf{y} = W_{opt}^T \mathbf{x} \tag{5.30}$$

Then the testing image can be recognized as taken for one of the individuals in the training set by using classification algorithms of the users preference. The face recognition method is proposed and performed in Turk and Pentland (1991b) Turk and Pentland (1991a), and has been popular and broadly used. Later it is compared with the recognition by Fisherfaces in Belhumeur et al. (1997), and compared with the recognition by Laplacianfaces in He et al. (2005), which are described in the following sections.

As commented in Adini et al. (1997), the changes of illumination account for much of the variation from one image to another. Also, one of the drawbacks of recognition by Eigenfaces that is discussed in Belhumeur et al. (1997) lies in the fact that the subspace spanned by Eigenfaces not only maximizes the total inter-class variance but also the intra-class variances (which mostly caused by changing lighting conditions and postures). This leads to the discussion on the idea of throwing away the first three eigenvectors that correspond to the largest three eigenvalues, which are considered as the ones account for the variations caused by the changing lighting conditions. Such experiments have been done and compared to other recognition algorithms Belhumeur et al. (1997). However, such strategy lacks of theoretical support that indicates the first few eigenvectors correspond solely to the variations by lighting conditions. Although in Belhumeur et al. (1997), some experiments have shown better recognition results by throwing away the most significant components, its authors also comment that this strategy may cause the loss of information on the variances useful for discrimination.

5.2.2 Face Recognition by Fisherfaces

As mentioned by the previous section, the Eigenfaces generate a linear subspace that maximizes the total scatter variance. This also means that the intra-class variances, which are mainly caused by different lighting conditions and postures of a same individual, are maxi-

mized in the projection procedure. This may result in inaccurate recognition due to the fact that much of differences between images are caused by the changes of lighting conditions Adini et al. (1997). The authors of Belhumeur et al. (1997) proposed a recognition method by using the class specific linear method for dimensionality reduction, and derived a new set of "principal images" that account for the maximized ratio of the total inter-class variance and the total intra-class variance. To realize this, this algorithm requires the knowledge of the classifications of the images in the training set. The class specific linear method used in their algorithm is Fisher's Linear Discriminant (FDA)R.A.Fisher (1936), and the optimization procedure can be summarized as follows.

Let $\{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\} \subset \mathbb{R}^n$ be the training set of images that are classified into c different groups corresponding to c different individuals. If we denote the mean vector of the training set as μ, and the mean vector of the i-th group as $\mu_{\mathbf{i}}$, then the total inter-class variance (between-class variance) of the training set can be computed as

$$S_B = \sum_{i=1}^{c} N_i(\mu_{\mathbf{i}} - \mu)(\mu_{\mathbf{i}} - \mu)^T \tag{5.31}$$

where N_i is the number of images in the training set that correspond to the i-th individual. The total intra-class variance (within-class variance) can be written as

$$S_W = \sum_{i=1}^{c} \sum_{\mathbf{x_k} \in G_i} (\mathbf{x_k} - \mu_{\mathbf{i}})(\mathbf{x_k} - \mu_{\mathbf{i}})^T \tag{5.32}$$

where G_i is the i-th group of images that correspond to the i-th individual.

Assume we are searching for a linear transformation mapping from the original images to a lower dimensional linear subspace (with dimensionality m) and let us write such a mapping as $\mathbf{y} = W^T\mathbf{x}$, then the optimal mapping currently under consideration is defined as the following

$$W_{opt} = \arg\max_W \frac{|W^T S_B W|}{|W^T S_W W|} = [\mathbf{w}_1 \ \mathbf{w}_2 \cdots \mathbf{w}_m] \tag{5.33}$$

whose columns $\mathbf{w}_1 \ \mathbf{w}_2 \cdots \mathbf{w}_m$ are the generalized eigenvectors of S_B and S_W corresponding to the m largest eigenvalues $\lambda_1, \lambda_2, \cdots, \lambda_m$. That is, $\mathbf{w}_i$ and λ_i satisfy the following equation

$$S_B\mathbf{w}_i = \lambda_i S_W \mathbf{w}_i, \quad i = 1, 2, \cdots, m \tag{5.34}$$

It is indicated in Duda and Hart (1973) Belhumeur et al. (1997) that there are at most $c-1$ nonzero generalized eigenvalues. Thus, the dimensionality m is at most $c-1$.

One problem of applying the above optimization is that the intra-class scatter matrix $S_W \in \mathbb{R}^{n\times n}$ is always singular due to the fact that the rank of S_W is at most $N-c$ and normally the number of training images N is much smaller than the dimensionality n of the images. Thus, it is proposed in Belhumeur et al. (1997) that the training set is firstly projected to a lower dimensional space (of dimensionality $N-c$) by performing PCA such that S_W is not singular anymore. Then, FDA is performed on this lower dimensional space to further reduce the dimensionality to $c-1$.

Thus, the algorithm computes W_{opt} as the following

$$W_{opt}^T = W_{fld}^T W_{pca}^T \tag{5.35}$$

where

$$W_{pca} = \arg\max_W |W^T S_T W| \tag{5.36}$$

and

$$W_{fld} = \arg\max_W \frac{|W^T W_{pca}^T S_B W_{pca} W|}{|W^T W_{pca}^T S_W W_{pca} W|} \tag{5.37}$$

Meanwhile, other alternatives can be taken to reduce the intra-class scatter and maximize the inter-class scatter. Another approach that proposed in Belhumeur et al. (1997) is to do optimization under a special constraint:

$$W_{opt} = \arg\max_{W\in\mathfrak{W}} |W^T S_B W| \tag{5.38}$$

where $\mathfrak{W}$ is the set of $n\times m$ matrices with orthonormal columns and contained in the kernel of S_W.

Experiments done in Belhumeur et al. (1997) show that the recognition by Fisherfaces applied on the image dataset obtained in the Harvard Robotics Laboratory outperforms other algorithms such as Eigenfaces (with and without the first three most significant components)

(see previous section), correlation method Brunelli and Poggio (1993) Gilbert and Yang (1993), linear subspace method Belhumeur et al. (1997). Other experiments shown in Belhumeur et al. (1997) also demonstrate the advantages of Fisherfaces applied on some other databases that contain variations in facial expression, eye wear, and lighting conditions.

5.2.3 Face Recognition by Laplacianfaces

Since there have been research showing that the face images taken under different lighting conditions and postures may reside on a nonlinear submanifold Chang et al. (2003b) Lee et al. (2003b) Roweis and Saul (2000b) Roweis et al. (2002b), a model that can capture such characteristic is necessary for more accurate face recognition. Although recognitions by Eigenfaces or Fisherfaces provide sufficient results on different databases, these two methods focus on the global structures of training sets by exploring the data structures in terms of Euclidean distances between images, which could be misleading if a more accurate distance defined on the nonlinear manifold is available. A model that attempts to preserve the local information and incorporates it into the global view may provide us a better face recognition algorithm without knowing the exact nonlinear submanifold. The face recognition algorithm by Laplacianfaces proposed in He et al. (2005) utilizes the local data structures of a given training set, extracts features of the latter by an optimization that preserves local similarities, then projects the testing images onto a lower dimensional subspace. This process is then followed by clustering or classification methods depending on the users' preferences or goals. The detailed procedures are described as follows.

Recognition by Laplacianfaces is based on the Locality Preserving Projection (LPP) He and Niyogi (2002) which learns a lower dimensional subspace while preserving the intrinsic geometry of the given dataset. Let the given dataset be $X = \{\mathbf{x}_i\}_{i=1}^n$, then the LPP procedure is realized through the optimization of the following:

$$\min \sum_{ij} (y_i - y_j)^2 S_{ij} \tag{5.39}$$

where y_i is the one-dimensional new representation of the data point $\mathbf{x}_i$, which usually lies in

a high dimensional space. S_{ij} is the similarity measure between $\mathbf{x}_i$ and $\mathbf{x}_j$, and should be one that preserves local data structure. One possibility is to define S_{ij} as follows:

$$S_{ij} = \begin{cases} \exp\left(-||\mathbf{x}_i - \mathbf{x}_j||^2/t\right), & \text{if } ||\mathbf{x}_i - \mathbf{x}_j|| < \epsilon \\ 0 & \text{else} \end{cases} \tag{5.40}$$

where ϵ is chosen by the user and should sufficiently small to capture the local properties.

Or S_{ij} can be defined using k nearest neighborhood:

$$S_{ij} = \begin{cases} \exp\left(-||\mathbf{x}_i - \mathbf{x}_j||^2/t\right), & \text{if } \mathbf{x}_i \in N_k(\mathbf{x}_j) \text{ or } \mathbf{x}_j \in N_k(\mathbf{x}_i) \\ 0 & \text{else} \end{cases} \tag{5.41}$$

where $N_k(\mathbf{x}_i) = \{\mathbf{x} \in X \mid \mathbf{x} \text{ is among the k nearest neighbors of } \mathbf{x}_i\}$. k is decided by the user and usually the nearest neighbors are found by using Euclidean distances.

Since the minimization given in (5.39) searches for new representation y_i for $\mathbf{x}_i$ ($i = 1, 2, \cdots, n$), it penalizes y_i and y_j for being far apart if the similarity S_{ij} is large. That is, the similarity between $\mathbf{x}_i$ and $\mathbf{x}_j$ leads to the similarity between y_i and y_j. Thus, the local structure is preserved through an appropriately defined similarity matrix S.

Assume that y_i is obtained by performing a linear operation on $\mathbf{x}_i$, that is , $y_i = \mathbf{w}^T\mathbf{x}_i$. The following computation shows the relationship between (5.39) and the Graph Laplacian:

$$\begin{aligned} \frac{1}{2}\sum_{ij}(y_i - y_j)S_{ij} &= \frac{1}{2}\sum_{ij}(\mathbf{w}^T\mathbf{x}_i - \mathbf{w}^T\mathbf{x}_j)^2 S_{ij} \\ &= \sum_{ij}\mathbf{w}^T\mathbf{x}_i S_{ij}\mathbf{x}_i^T\mathbf{w} - \mathbf{w}^T\mathbf{x}_i S_{ij}\mathbf{x}_j^T\mathbf{w} \\ &= \sum_{i}\mathbf{w}^T\mathbf{x}_i D_{ii}\mathbf{x}_i^T\mathbf{w} - \mathbf{w}^T X S X^T\mathbf{w} \\ &= \mathbf{w}^T X(D - S)X^T\mathbf{w} \\ &= \mathbf{w}^T X L X^T\mathbf{w} \end{aligned}$$

where the degree matrix D is a diagonal matrix with $D_{ii} = \sum_j S_{ij}, i = 1, 2, \cdots, n$. $L = D - S$ is the Graph Laplacian matrix. Since D_{ii} can be considered as a measure of "importance" of the data point $\mathbf{x}_i$, a constraint can be posed to the minimization as follows:

$$\mathbf{y}^T D\mathbf{y} = 1 \quad \Rightarrow \quad \mathbf{w}^T XDX^T\mathbf{w} = 1 \tag{5.42}$$

Thus the minimization can be rewritten as

$$\arg \min_{\mathbf{w}^T XDX^T\mathbf{w}} \mathbf{w}^T XLX^T\mathbf{w} \tag{5.43}$$

And the minimizer $\mathbf{w}$ can be found as the eigenvector that corresponds to the smallest eigenvalue of the following generalized eigenvalue problem

$$XLX^T\mathbf{w} = \lambda XDX^T\mathbf{w} \tag{5.44}$$

The Graph Laplacian can be considered as the discrete approximation of the Laplace-Beltrami operator that is defined on the nonlinear submanifold, and the eigenvectors of the former are approximations of the eigenfunctions of the latter under some conditions as discussed in the previous chapters. Since the eigenfunctions of the Laplace Beltrami operator can capture the structure of the manifold that the dataset resides on, so can the eigenvectors of the Graph Laplacian in the approximation sense.

The above described LPP is a general method for manifold learning by approximating the eigenfunctions of the Laplace Beltrami operator. To derive a lower dimensional representation (of dimensionality k) of the given dataset, we can use the first k eigenvectors of (5.44) that correspond to the smallest k eigenvalues. Then these eigenvectors form the basis for the desired lower dimensional subspace. Although LPP gives a linear subspace as the result of dimensionality reduction, it is a locally topology-preserving mapping and is able to encode the local data structure on the nonlinear manifold in the linear subspace.

LPP can be used in the dimensionality reduction process of the face recognition algorithm. But one problem of solving (5.44) directly is that the matrix XDX^T is usually singular He et al.

(2005). Thus, a PCA process is applied first to the training set in order to solve this problem, and at the same time, reduce the noise. We can denote such a procedure using the mapping $\mathbf{x} \to W_{PCA}^T \mathbf{x}$, where W_{PCA} is the projection matrix with the first several significant components of $X^T X$ as its columns. Then, both $X^T L X$ and $X D^T X$ are symmetric and positive semi-definite, and the solutions of (5.44) can be denoted as $\mathbf{w}_0, \mathbf{w}_1, \cdots, \mathbf{w}_{k-1}$ corresponding to eigenvalues $0 \leq \lambda_0 \leq \lambda_1 \leq \cdots \leq \lambda_{k-1}$.

In all, the mapping between the original training set X to the lower dimensional subspace that is derived from PCA and LPP is:

$$\mathbf{x} \to \mathbf{y} = W^T \mathbf{x} \tag{5.45}$$

$$W = W_{PCA} W_{LPP} \tag{5.46}$$

where $W_{LPP} = [\mathbf{w}_0, \mathbf{w}_1, \cdots, \mathbf{w}_{k-1}]$. The columns of W are called Laplacianfaces.

The recognition algorithm by Laplacianfaces are applied on the Yale dataset and the CMU PIE face database respectively, compared with the algorithms associated with Eigenfaces and Fisherfaces He et al. (2005). Both of the datasets consist of images taken for different individual under varying pose, illumination conditions, and facial expressions. Part of the dataset is used as the training set for both datasets, and Laplacianfaces, Eigenfaces, Fisherfaces are used in the dimensionality reduction procedures, respectively. For each dataset, the best recognition results obtained from these three procedures are compared. It is shown in He et al. (2005) that the Laplacianfaces achieves better recognition result compared to the other two methods when they all reach their best performances on a range of values of the dimension k.

5.3 Incorporating Fuzzy-RW Into Face Recognition Algorithms

In this section, we compare the performances of FCM, spectral clustering method, Fuzzy-RW by applying them in solving the face recognition problem. We consider the Yale dataset.

The Yale data base consists of 165 images of 15 individuals. Each individual has 11 images taken with different facial expressions or under different lighting conditions. In our experiment, 5 out of 11 images per individual were taken to form a training set from which lower dimensional representatives of the Yale images are found through eigenfaces or laplacianfaces techniques. Fig.5.1 shows the clustering results derived by applying FCM, spectral clustering method, Fuzzy-RW with Eigenfaces or Laplacianfaces. It is clear that Fuzzy-RW outperforms the other clustering methods, and it correctly recognizes the 15 groups with excellent precision.

The "bandwidth" parameter σ that is used in defining the weight matrix:

$$W = (W_{ij})_{N\times N}, \quad W_{ij} = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{\kappa^\gamma \sigma}\right)$$

has a fundamental influence on the clustering results. Here in the face recognition context, we discuss a possible method of finding an appropriate "bandwidth" using a chosen training set.

Let X be the given dataset, and T be the chosen training set from which we will learn an appropriate "bandwidth" parameter. Since the size of a training set is comparatively small, a manual face recognition can be achieved, and denoted as a column vector $R_T = (R_T^i)_{n_T\times 1}$, where $n_T = |T|$ and $R_T^i =$ the index of the individual that the ith image was taken from.

To search for an appropriate "bandwidth" parameter that generates the best approximation of R_T among a range of values, we can run experiments using a sequence of σ_h, which is defined as

$$\sigma_h = (d_{\min} + h \cdot [d_{\max} - d_{\min}])^2 \tag{5.47}$$

where $d_{\min} = \min\{||\mathbf{x}_i - \mathbf{x}_j|| \mid \mathbf{x}_i \in X, \mathbf{x}_j \in X\}$ and $d_{\max} = \max\{||\mathbf{x}_i - \mathbf{x}_j|| \mid \mathbf{x}_i \in X, \mathbf{x}_j \in X\}$, and $h \in [0, 1]$. The clustering result corresponding to σ_h can be represented as a column vector $R_h = (R_h^i)_{n_T\times 1}$ and $R_h^i =$ the index of the individual to which the ith image is clustered by using σ_h in the weight matrix.

Then one possible method of finding the best σ_h when $h \in H$ with $H \subset [0,1]$ is to find the following

$$h^* = \arg\min_{h \in H} \|R_h - R_T\|_{l^2} \tag{5.48}$$

Then $\sigma_{h^*} = (d_{\min} + h^* \cdot [d_{\max} - d_{\min}])^2$ learnt from the training set T can be considered as an appropriate "bandwidth" parameter for clustering the whole dataset.

As a demonstration of this learning process, let us consider a training set T as the subset of the Yale Database that consists of the images taken when each individual had 4 different face expressions ("normal", "sleepy", "sad", "happy") and the images taken when the lighting condition is "central light". That is, this training set consists of 5 images taken for each individual under various conditions, and there are 75 training images in total. We can take H as $\{h = 0.05 \cdot i,\ i = 0, 1, 2, \cdots, 20\}$. For this training set T, we used the Eigenfaces method for the dimensionality reduction, and applied Fuzzy-RW with the absorption distance to derive the clustering results for each $h \in H$, then the best "bandwidth" parameter σ_{h*} in the range H was found as approximately 341. Fig. 5.2(a) shows the clustering results on the training set T using σ_{h^*}. Using σ_{h^*} as the "bandwidth" parameter to compute the absorption distance, which is then used in Fuzzy-RW to cluster the whole dataset, we generated the face recognition results shown in Fig. 5.2(b). The following discusses the details of these experiments. In these experiments, the weight matrix has the following form:

$$W_{ij} = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{\sigma}\right)$$

The absorption distances are defined as in the previous chapter:

$$d(\mathbf{x}_i, \mathbf{x}_j) = [1 - 1/2(P_i(\tau_j < \tau_i^R) + P_j(\tau_i < \tau_j^R))]^\beta$$

The results shown in Fig. 5.2 are generated with $\beta = 3$. In the objective function of Fuzzy-RW, the controlling parameter K is set to be N/C_n^2, where N is the number of images and n is the number of clusters. From the results shown in these figures, it can be concluded that the above process can serve as a systematic method of finding an appropriate "bandwidth" parameter based on a given training set.

Another set of experiments are shown below to explore the influence of different training sets in the face recognition results. For example, in the following experiments we choose the training sets $T_1 \subset T_2$ with $|T_1| = 90$, and $|T_2| = 135$. T_1 consists the images taken for each individual when they have face expressions "normal", "happy", "sleepy", "sad", and the lighting condition is "centerlight", and when they wear no glasses. T_2 expands T_1 by adding the images taken for each individual when they wear glasses, and when the lighting conditions are set to be "leftlight" and "rightlight".

Here we demonstrate the face recognition results in two experiments, these experiments use T_1 and T_2 respectively as the training set to derive the corresponding Eigenfaces. Then the clustering method, Fuzzy-RW with commute distances, is applied to generate the face recognition results on the test sets. Let X be the whole Yale Database, then the test sets that correspond to T_1, T_2 are $X_1 = X \backslash T_1$, $X_2 = X \backslash T_2$ respectively. It is expected that by enlarging the training set, more detailed information that benefits the face recognition result is included in the corresponding Eigenfaces, thus we should derive more accurate results. Fig.5.3 shows the face recognition results on X_1, X_2 by using the Eigenfaces derived from T_1, T_2 respectively. For each of the two experiments, a best "bandwidth" parameters were chosen from a range of values. Let $d_{\min} = \min\{\|\mathbf{x}_i - \mathbf{x}_j\| \mid \mathbf{x}_i \in X, \mathbf{x}_j \in X\}$, and $d_{\max} = \max\{\|\mathbf{x}_i - \mathbf{x}_j\| \mid \mathbf{x}_i \in X, \mathbf{x}_j \in X\}$. The "bandwidth" parameter that used to generate Fig.5.3 (a) is $[d_{\min} + 1/12(d_{\max} - d_{\min})]^2$ and the one used in Fig.5.3 (b) is $[d_{\min} + 1/10(d_{\max} - d_{\min})]^2$. The percentages of correctly clustered images in the two test sets are 61.3% and 77%, respectively. The penalty parameter used for Fig.5.3 (a) is $K = 10^{38}$ and the one used for Fig.5.3 (b) is $K = 10^{45}$.

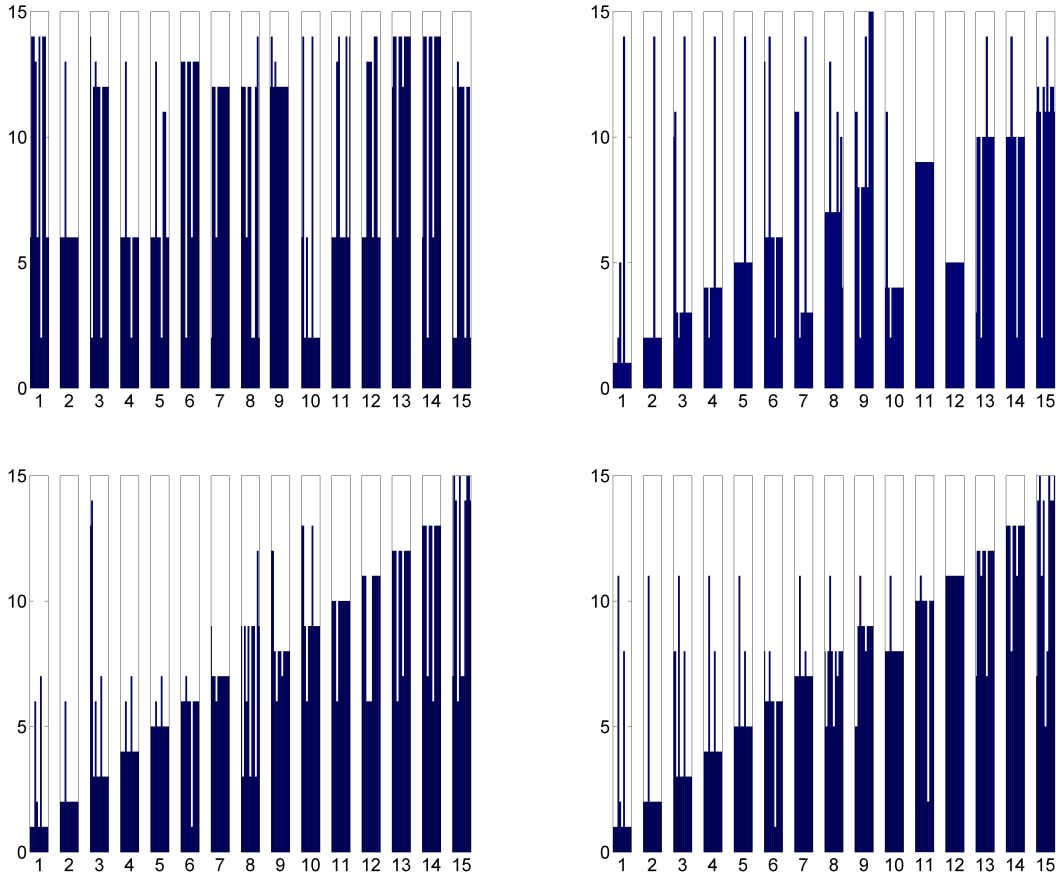


Figure 5.1: Training set consists of 75 images. From (a) to (c), the dimensionality reduction is done by using the eigenface technology. (a). Result given by FCM with the number of clusters set to be 15. (b). Result given by spectral clustering with weight matrix defined as $W_{ij} = \exp\left(-||\mathbf{x}_i - \mathbf{x}_j||^2/\sigma\right)$ with $\sigma = 20000$. K-Means method is applied on the eigenvectors that correspond to the smallest 15 nonzero eigenvalues of the graph Laplacian. (c). Result given by Fuzzy-RW with commute distance, $\sigma = 127.6$, $r = 22.6$, $s = 2$, $\gamma = 1/6$, and $K = 10^{38}$. (d). Result given by first reducing the dimensionality using the Laplacianfaces technology then applying Fuzzy-RW with commute distance. The parameters used in Fuzzy-RW are $\sigma = 51$, $\gamma = 0$ and $K = 10^{40}$.

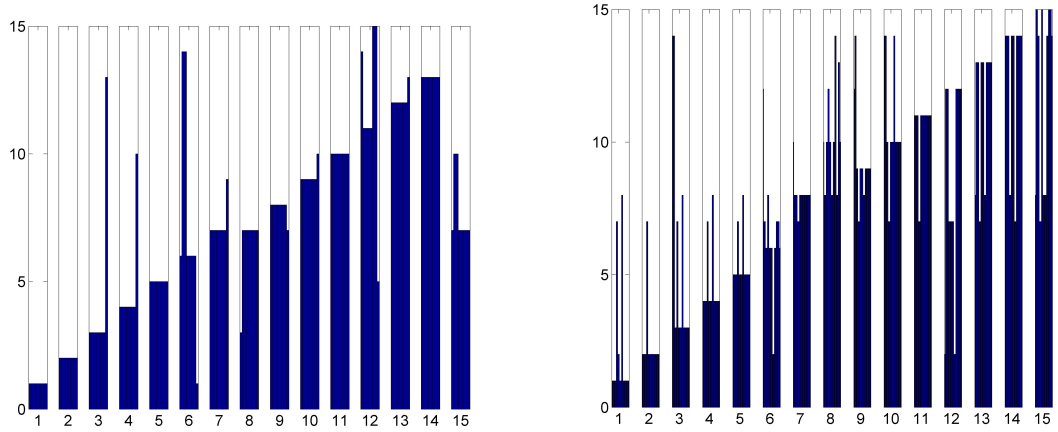


Figure 5.2: (a). The clustering result on the training set T. This result is generated by using σ_{h*} learnt from T, and among the range H. (See the text for details.) (b). The face recognition results of the whole Yale Database using Fuzzy-RW with σ_{h*} and absorption distances. (See text for details of parameters.)

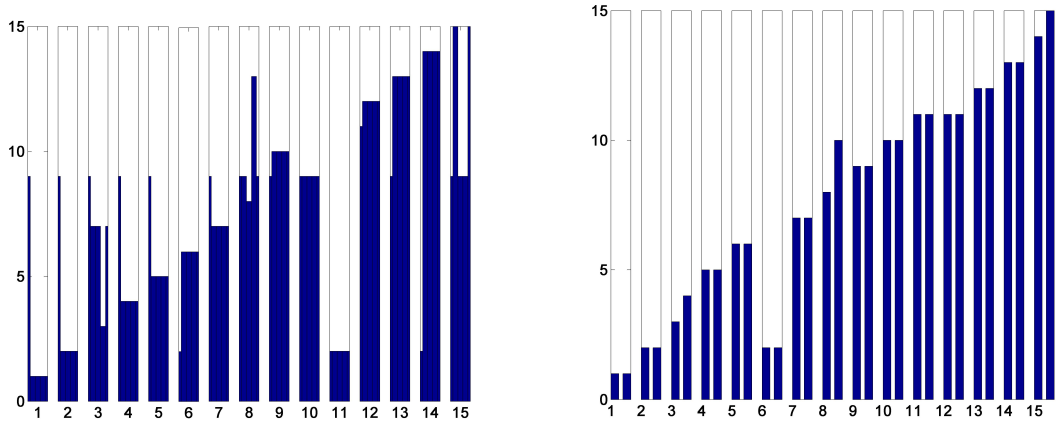


Figure 5.3: (a). Face recognition results on the test set $X_1 = X \backslash T_1$. (b). Face recognition results on the test set $X_2 = X \backslash T_2$. (See text for details of relative parameters.)

BIBLIOGRAPHY

Adini, Y., Moses, Y., and Ullman, S. (1997). Face recognition: the problem of compensating for changes in illumination direction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19:721–732.

Ahmed, M., Yamany, S., Mohamed, N., Farag, A., and Moriarty, T. (2002). A modified fuzzy c-means algorithm for bias field estimation and segmentation of mri data. *Medical Imaging, IEEE Transactions on*, 21(3):193 –199.

Archip, N., Rohling, R., Cooperberg, P., Tahmasebpour, H., and Warfield, S. (2005). Spectral clustering algorithms for ultrasound image segmentation. In Duncan, J. and Gerig, G., editors, *Medical Image Computing and Computer-Assisted Intervention MICCAI 2005*, pages 862–869. Springer.

Bach, F. R. and Jordan, M. I. (2004). Learning Spectral Clustering. In *Advances in Neural Information Processing Systems 16*. MIT Press, Cambridge, MA.

Ball, G. H. and Hall, D. J. (1965). ISODATA. A novel method of data analysis and pattern classification. Technical report, Menlo Park: Stanford Research Institute.

Banerjee, A., Krumpelman, C., and Ghosh, J. (2005a). Model-based overlapping clustering. In *In KDD*, pages 532–537. ACM Press.

Banerjee, A., Merugu, S., Dhillon, I., and Ghosh, J. (2005b). Clustering with bregman divergences. In *JOURNAL OF MACHINE LEARNING RESEARCH*, pages 1705–1749. JMLR.org.

Basri, R. and Jacobs, D. (2003). Lambertian reflectance and linear subspaces. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 25(2):218 – 233.

Belhumeur, P. and Kriegman, D. (1996). What is the set of images of an object under all possible lighting conditions? In *Computer Vision and Pattern Recognition, 1996. Proceedings CVPR '96, 1996 IEEE Computer Society Conference on*, pages 270 –277.

Belhumeur, P. N., Hespanha, J. P., and Kriegman, D. J. (1997). Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):711–720.

Belkin, M. and Niyogi, P. (2003). Laplacian Eigenmaps for dimensionality reduction and data representation. *Neural Comput.*, 15(6):1373–1396.

Belkin, M. and Niyogi, P. (2005). Towards a theoretical foundation for laplacian-based manifold methods. pages 486–500. Springer.

Bezdek, J., Ehrlich, R., and Full, W. (1984). FCM: The fuzzy c-means clustering algorithm. *Computers & Geosciences*, 10:191–203.

Bezdek, J. C. (1981). *Pattern Recognition with Fuzzy Objective Function Algorithms (Advanced Applications in Pattern Recognition)*. Springer.

Bezdek, J. C., Keller, J., Krisnapuram, R., and Pal, N. (2005). *Fuzzy Models and Algorithms for Pattern Recognition and Image Processing (The Handbooks of Fuzzy Sets)*. Springer.

Bie, T. D., Cristianini, N., Bennett, P., and Parrado-hernndez, E. (2006). Fast sdp relaxations of graph cut clustering, transduction, and other combinatorial problems. *JMLR*, 7:1409–1436.

Binford, T. O. (1988). Generic surface interpretation: observability model. In *Proceedings of the 4th international symposium on Robotics Research*, pages 265–272, Cambridge, MA, USA. MIT Press.

Bousquet, O., Chapelle, O., and Hein, M. (2004). Measure based regularization. In *Advances in Neural Information Processing Systems 16*. MIT Press.

Bronstein, A., Bronstein, M., and Kimmel, R. (2008). *Numerical Geometry of Non-Rigid Shapes*. Springer.

Brunelli, R. and Poggio, T. (1993). Face recognition: features versus templates. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 15(10):1042 –1052.

Bui, T. N. and Jones, C. (1992). Finding good approximate vertex and edge partitions is np-hard. *Information Processing Letters*, 42(3):153 – 159.

Celleux, G. and Govaert, G. (1992). A classification EM algorithm for clustering and two stochastic versions. In *Computational Statistics and Data Analysis*, volume 14, pages 315–332.

Chan, P. K., Schlag, M. D. F., and Zien, J. Y. (1993). Spectral k-way ratio-cut partitioning and clustering. In *DAC '93: Proceedings of the 30th international conference on Design automation*. ACM.

Chang, Y., Hu, C., and Turk, M. (2003a). Manifold of Facial Expression. In *In IEEE International Workshop on Analysis and Modeling of Faces and Gestures*, pages 28–35.

Chang, Y., Hu, C., and Turk, M. (2003b). Manifold of facial expression. *Analysis and Modeling of Faces and Gestures, IEEE International Workshop on*, 0:28.

chih Lee, K., Ho, J., and Kriegman, D. (2005). Acquiring linear subspaces for face recognition under variable lighting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27:684–698.

Chung, F. R. K. (1997). *Spectral Graph Theory (CBMS Regional Conference Series in Mathematics, No. 92)*. American Mathematical Society.

Cleuziou, G. (2008). An extended version of the k-means method for overlapping clustering. In *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, pages 1 –4.

Cleuziou, G. (2010). Two variants of the okm for overlapping clustering. In *Advances in Knowledge Discovery and Management*, volume 292 of *Studies in Computational Intelligence*, pages 149–166. Springer Berlin / Heidelberg.

Coifman, R. and Lafon, S. (2006). Diffusion maps. *Appl. Comput. Harmon. Anal.*, 21:5–30.

Cominetti, O., Matzavinos, A., Samarasinghe, S., Kulasiri, D., Liu, S., Maini, P., and Erban, R. (2010). Diffuzzy: A fuzzy clustering algorithm for complex data sets. *Int. J. Computational Intelligence in Bioinformatics and Systems Biology*, 1(4):402 – 417.

Cour, T., Benezit, F., and Shi, J. (2005a). Spectral Segmentation with Multiscale Graph Decomposition. In *CVPR '05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 2*, Washington, DC, USA. IEEE Computer Society.

Cour, T., Gogin, N., and Shi, J. (2005b). Learning Spectral Graph Segmentation. In *IEEE International Conference on Artificial Intelligence and Statistics*.

Cour, T. and Shi, J. (2004). A learnable spectral memory graph for recognition and segmentation.

Dattola, R. (1968). A fast algorithm for automatic classification. Technical report, Report ISR-14 to the National Science Foundation, Section V, Cornell University, Department of Computer Science.

del Solar, J. R. and Navarrete, P. (2005). Eigenspace-based face recognition: a comparative study of different approaches. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 35(3):315 –325.

Diday, E. (1987). Orders and overlapping clusters by pyramids. Technical report, INRIA num.730, Rocquencourt 78150, France.

Ding, C. and He, X. (2004). K-means clustering via principal component analysis. pages 225–232. ACM Press.

Drineas, P., Frieze, A., Kannan, R., Vempala, S., and Vinay, V. (2004). Clustering large graphs via the singular value decomposition. In *MACHINE LEARNING*, pages 9–33.

Duda, R. and Hart, P. (1973). *Pattern Classification and Scene Analysis*. New York: Wiley.

Dunn, J. C. (1973). A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters. *Journal of Cybernetics*, 3(3):32–57.

Enright, A. J., Van Dongen, S., and Ouzounis, C. A. (2002). An efficient algorithm for large-scale detection of protein families. *Nucleic Acids Research*, 30(7):1575–1584.

Epstein, R., Hallinan, P., and Yuille, A. (1995). 5 plus minus 2 eigenimages suffice: an empirical investigation of low-dimensional lighting models. In *Physics-Based Modeling in Computer Vision, 1995., Proceedings of the Workshop on*, page 108.

Fan, K. (1949). On a theorem of Weyl concerning eigenvalues of linear transformations. I. *Proc. Nat. Acad. Sci. U. S. A.*, 35:652–655.

Filippone, M., Camastra, F., Masulli, F., and Rovetta, S. (2007). A survey of kernel and spectral methods for clustering. *Pattern Recognition*, 41:176–190.

Fouss, F., Pirotte, A., michel Renders, J., and Saerens, M. (2006). Random-walk computation of similarities between nodes of a graph, with application to collaborative recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 19:2007.

Fu, L. and Medico, E. (2007). FLAME, a novel fuzzy clustering method for the analysis of DNA microarray data. *BMC Bioinformatics*, 8(3).

Georghiades, A., Belhumeur, P., and Kriegman, D. (2000). From few to many: generative models for recognition under variable pose and illumination. In *Automatic Face and Gesture Recognition, 2000. Proceedings. Fourth IEEE International Conference on*, pages 277 –284.

Georghiades, A. S., Kriegman, D. J., and Belhumeur, P. N. (1998). Illumination cones for recognition under variable lighting: Faces. pages 52–59.

Gilbert, J. M. and Yang, W. (1993). A real-time face recognition system using custom vlsi hardware.

Golub, G. H. and Van Loan, C. F. (1989). *Matrix computations* . Johns Hopkins University Press.

Gutman, I. and Xiao, W. (2004). Generalized inverse of the laplacian matrix and some applications. *Bull. Acad. Serb. Sci. Arts (Cl. Math. Natur.)*, 129:15 – 23.

Hagen, L., K. A. (1992). New spectral methods for ratio cut partitioning and clustering. *IEEE Trans. Comput.-Aided Des.*, 11.

Hallinan, P. (1994). A low-dimensional representation of human faces for arbitrary lighting conditions. In *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR '94., 1994 IEEE Computer Society Conference on*, pages 995 –999.

Hathaway, R. and Bezdek, J. (2001). Fuzzy *c*-means clustering of incomplete data. *IEEE Transactions on Systems, Man, and Cybernetics – Part B*, 31(5):735–744.

He, X. and Niyogi, P. (2002). Locality preserving projections.

He, X., Yan, S., Hu, Y., Niyogi, P., and jiang Zhang, H. (2005). Face recognition using laplacianfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27:328–340.

Higham, D. J., Kalna, G., and Kibble, M. (2007). Spectral clustering and its use in bioinformatics.

Hoffmann, H. (2007). Kernel pca for novelty detection. *Pattern Recognition*, 40.

Jain, A. K. (2008). Data clustering: 50 years beyond k-means.

Jardine, N. and Sibson, R. (1971). *Mathematical Taxonomy.* John Wiley and Sons Ltd., London.

Jean-Philippe Vert, K. T. and Scholkopf, B. (2004). A primer on kernel methods. In *Kernel Methods in Computational Biology*, pages 35–70. MIT Press.

Kannan, R., Vempala, S., and Vetta, A. (2000). On clusterings: Good, bad and spectral.

Kogan, J. (2007). *Introduction to Clustering Large and High-Dimensional Data.* Cambridge University Press.

Lee, K.-C., Ho, J., Yang, M.-H., and Kriegman, D. (2003a). Video-based face recognition using probabilistic appearance manifolds. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 1, pages I–313–I–320.

Lee, K.-C., Ho, J., Yang, M.-H., and Kriegman, D. (2003b). Video-based face recognition using probabilistic appearance manifolds. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 1, pages I–313 – I–320 vol.1.

Liao, C.-S., Lu, K., Baym, M., Singh, R., and Berger, B. (2009). IsoRankN: spectral methods for global alignment of multiple protein networks. *Bioinformatics*, 25:i253–i258.

Liew, A., Leung, S., and Lau, W. (2000). Fuzzy image clustering incorporating spatial continuity. *Vision, Image and Signal Processing, IEE Proceedings* -, 147(2):185 –192.

Linde, Y., Buzo, A., and Gray, R. (1980). An algorithm for vector quantizer design. *Communications, IEEE Transactions on*, 28(1):84 – 95.

Lloyd, S. P. (1982). Least squares quantization in pcm. *IEEE Transactions on Information Theory*, 28:129–137.

Luxburg, U. (2007). A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416.

Ma;, G. G. C. and Wu, J. (2007). *Data Clustering: Theory, Algorithms, and Applications (ASA-SIAM Series on Statistics and Applied Probability)*. SIAM, Society for Industrial and Applied Mathematics.

MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. In *Proc. Fifth Berkeley Symp. on Math. Statist. and Prob.*, volume 1, pages 281–297. Univ. of Calif. Press.

Malik, J. and Perona, P. (1990). Preattentive texture discrimination with early vision mechanisms. *Journal of the Optical Society of America A*, 7:923–932.

Mao, J. and Jain, A. (1996). A self-organizing network for hyperellipsoidal clustering (hec). *Neural Networks, IEEE Transactions on*, 7(1):16 –29.

Meila, M. (2006). The uniqueness of a good optimum for k-means. In *Proc. 23rd Internat. Conf. Machine Learning*, pages 625–632.

Meila, M. and Shi, J. (2000). Learning Segmentation by Random Walks. In *NIPS*, pages 873–879.

Moghaddam, B. (2002). Principal manifolds and probabilistic subspaces for visual recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(6):780 –788.

Mohar, B. (1991). The laplacian spectrum of graphs. In *Graph Theory, Combinatorics, and Applications*, pages 871–898. Wiley.

Mohar, B. and Juvan, N. T. M. (1997). Some applications of laplace eigenvalues of graphs. In *Graph Symmetry: Algebraic Methods and Applications, volume 497 of NATO ASI Series C*, pages 227–275. Kluwer.

Nash, J. (1954). C^1 Isometric Imbeddings. *Annals of Mathematics*, 56:383–396.

Ng, A. Y., Jordan, M. I., and Weiss, Y. (2001). On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems*, pages 849–856. MIT Press.

Paccanaro, A., Chennubhotla, C., Casbon, J., and Saqi, M. (2003). Spectral clustering of protein sequences. In *Neural Networks, 2003. Proceedings of the International Joint Conference on*, volume 4, pages 3083 – 3088.

qiang Zhang, D. and can Chen, S. (2004). A novel kernelized fuzzy c-means algorithm with application in medical image segmentation. *Artificial Intelligence in Medicine*, 32:2004.

R.A.Fisher (1936). The use of multiple measures in taxonomic problems. *Ann. Eugenics*, 7:179–188.

Rosenberg, S. (1997). *The Laplacian on a Riemannian manifold*. Cambridge: Cambridge University Press.

Roweis, S., Saul, L. K., and Hinton, G. E. (2002a). Global coordination of local linear models. In *Advances in Neural Information Processing Systems 14*, pages 889–896. MIT Press.

Roweis, S., Saul, L. K., and Hinton, G. E. (2002b). Global coordination of local linear models. In *Advances in Neural Information Processing Systems 14*, pages 889–896. MIT Press.

Roweis, S. T. and Saul, L. K. (2000a). Nonlinear Dimensionality Reduction by Locally Linear Embedding. *Science*, 290:2323–2326.

Roweis, S. T. and Saul, L. K. (2000b). Nonlinear dimensionality reduction by locally linear embedding. *SCIENCE*, 290:2323–2326.

Seung, S. H. and Lee, D. D. (2000). The manifold ways of perception. *Science (New York, N.Y.)*, 290:2268–2269.

Shashua, A., Levin, A., and Avidan, S. (2002). Manifold pursuit: A new approach to appearance based recognition. In *International Conference On Pattern Recognition (ICPR*, pages 590–594.

Shi, J. and Malik, J. (1998). Motion segmentation and tracking using normalized cuts. In *in International Conference on Computer Vision*, pages 1154–1160.

Shi, J. and Malik, J. (2000). Normalized Cuts and Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905.

Singer, A. (2006). From graph to manifold laplacian: The convergence rate. *Appl. Comput. Harmon. Anal.*, 21:128–134.

Snel, B., Bork, P., and Huynen, M. (2002). The identification of functional modules from the genomic association of genes. *PNAS*, 99(9):5890 5895.

Spielman, D., T. S. (1996). Spectral clustering works: planar graphs and finite element meshes. *IEEE Comput. Soc. Press, Los Alamitos.*

Steinhaus, H. (1956). Sur la division des corp materiels en parties. *Bull. Acad. Polon. Sci*, 1:801–804.

Stewart, G. W. and Sun, J.-G. (1990). *Matrix Perturbation Theory (Computer Science and Scientific Computing).* Academic Press.

Stoer, M. and Wagner, F. (1997). A simple min-cut algorithm. *Journal of the ACM*, 44(4):585–591.

Tenenbaum, J. B., Silva, V., and Langford, J. C. (2000). A Global Geometric Framework for Nonlinear Dimensionality Reduction. *Science*, 290(5500):2319–2323.

Turk, M. and Pentland, A. (1991a). Eigenfaces for Recognition. *Journal of Cognitive Neuroscience*, 3(1):71–86.

Turk, M. A. and Pentland, A. P. (1991b). Face recognition using eigenfaces. In *Proceedings. 1991 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 586–591. IEEE Comput. Sco. Press.

Wardetzky, M., Mathur, S., Kalberer, F., and Grinspun, E. (2007). Discrete laplace operators: No free lunch. *Proc. Symp. Geometry Processing (SGP)*, pages 33–37.

Wu, Z. and Leahy, R. (2002). An optimal graph theoretic approach to data clustering: theory and its application to image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 15.

Yen, D., Vanvyve, F., Wouters, F., Fouss, F., Verleysen, M., and Saerens, M. (2005). Clustering using a random walk based distance measure. In *Proc. 13th European Symposium on Artificial Neural Networks (ESANN)*.

Yuille, A. L., Snow, D., Epstein, R., and Belhumeur, P. N. (1999). Determining generative models of objects under varying illumination: Shape and albedo from multiple images using svd and integrability. *International Journal of Computer Vision*, 35:203–222.

Zha, H., Ding, C., Gu, M., He, X., and Simon, H. (2001). Spectral relaxation for k-means clustering.

Zhang, D.-Q., Chen, S.-C., Pan, Z.-S., and Tan, K.-R. (2003). Kernel-based fuzzy clustering incorporating spatial constraints for image segmentation. In *Machine Learning and Cybernetics, 2003 International Conference on*, volume 4, pages 2189 – 2192 Vol.4.

PUBLICATION LIST

Peer Reviewed Journal Articles

1. K. Preedy, P.G. Schofield, **S. Liu**, A. Matzavinos, M. Chaplain, S.F. Hubbard, 2010, Modelling contact spread of infection in host-parasitoid systems: vertical transmission of pathogens can cause chaos. Journal of Theoretical Biology. **262**(3): 441-451.

   Reference [1] has been listed as one of the ScienceDirect Top 25 Hottest Articles for October 2009 - September 2010.

2. O. Cominetti, A. Matzavinos, S. Samarasinghe, D. Kulasiri, **S. Liu**, P.K. Maini, and R. Erban, 2010, DifFUZZY: A fuzzy clustering algorithm for complex data sets. International Journal of Computational Intelligence in Bioinformatics and Systems Biology. **1**(4): 402-417. [1]

3. **S. Liu**, A. Matzavinos, and S. Sethuraman. Random walk distances in data clustering and applications. *Submitted for publication.*

Papers in Preparation

4. A. Matzavinos, A. Roitershtein, Z. Voller, **S. Liu**, and M. Chaplain. On a stochastic model for a possible function of syntelic and merotelic kinetochores. *In preparation.*

[1] DifFUZZY has a dedicated website hosted online at the Centre for Mathematical Biology of the University of Oxford. The URL is `http://www.maths.ox.ac.uk/cmb/difFUZZY`